

要素マッピング手法による Rich Layout の実現

Implementation of Rich Layout by Element Mapping Method

日根埜谷 秀男[†] 中野 秀男[‡]

Hideo HINENOYA Hideo NAKANO

概要 ドキュメント中心(Document Centric) XML 文書の利用が意外と進んでいない理由として、XSLT の習得や構造化文書スキーマの理解が困難であるなどが挙げられるが、それ以上に問題なのは Rich なレイアウトへの変換である。その理由として XML 文書は「文書構造と表示」が分離しているため、Rich なレイアウトの実現には XSLT 作成に多大な負荷がかかりまたこのような XSLT は再利用が困難という問題も発生する。本論文ではこのような問題を解決するため、レイアウト情報・書式情報をマッピングファイルに記述し、このデータを XSLT 変換することで Rich なレイアウトと XSLT の再利用をある程度実現できた。

キーワード : XSLT, XSLT の再利用, XSL-FO, 構造化文書の変換, リッチレイアウト

Keywords : XSLT, Reuse of XSLT, XSL-FO, Transformation of Structured Document, Rich Layout

1. 始めに

近年、書籍や論文、法令集、約款、カタログ等さまざまな文書が XML(Extensible Markup Language)[1]化され、この XML データから WWW(World Wide Web)用として HTML(HyperText Markup Language)、紙媒体として PDF(Portable Document Format)に出力されることが多くなった。いわゆる、One Source, Multi Use である。データソースが単一であるため、データに変更が発生した場合、XML を変更すれば WWW も紙も最新情報に変更されるからである。またデータが大量かつ変換先フォーマットが多様であればあるほどこのメリットは大きい。

XML は文書の論理構造とレイアウト情報が分離されているので何らかの可視化が必要である。可視化の主な変換方法として WWW 環境では、

- (1) XML に CSS(Cascading Style Sheets)[2]を適用する。
- (2) XSLT(Extensible Stylesheet Language for Transformation)[3]により XML を HTML に変換する。

紙媒体では、

- (1) XML に CSS3(CSS Paged Media Level 3)[13]を適用して PDF 変換[14][15]。

- (2) 組版コマンド変換用 XSLT で LaTeX[6]等に変換。

- (3) XML を XSLT で XSL-FO(XSL Formatting Objects)[4]に変換後、XSL-FO プロセッサにより PDF、PS(Post Script)[5]や RTF(Rich Text Format)[8]などのフォーマット変換。

などが行われている。いずれの方法を採用しても鍵になるのはスタイルシートである。XML 文書にはテキスト本体と構造だけが書かれていて、その表示法つまりレイアウトに関する情報は含まれていない。したがって XML 文書を可視化する際には、出力用のアプリケーションに対してレイアウトを指示する必要がある。この指示書がスタイルシートである。XML 文書は特定のアプリケーションに依存しないというメリットがある。その反面、XML 文書を表示・印刷するには必ずスタイルシートを作成しなければならないという面倒さもある。商品カタログ・マニュアルなどの場合、ある程度 APPEARANCE (見栄え) が求められるため、Rich なレイアウトを実現するには XSLT 作成に多大な負担がかかる。特に「ページ概念 (本などの紙媒体ではテキストをページという単位で分割するが、ホームページなどのコンピュータ画面ではページという区切りは特に存在しない)」を持つ紙媒体への変換が問題である。

[†] 大阪市立大学大学院創造都市研究科都市情報学専攻 (情報基盤研究分野)

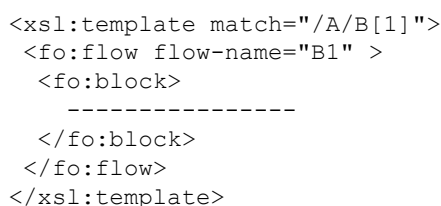
[‡] 大阪市立大学大学院創造都市研究科 兼 学術情報総合センター

本論文ではまず第3章でXMLデータ構造とそのXSLTの関係を、第4章でドキュメント中心XML文書のレイアウト方法を、第5章で再利用可能なXSLTと要素のマッピング例を述べる。また第6章ではマッピングファイルの概要とその検証方法を、第7章では具体的な変換ワークフローとテストケースを示す。最後に付録としてテストケースで使用したデータと出力例、並びにXSLTとマッピングファイル検証用スキマトロンスキーマを示す。なお、XSLTの出力をXSL-FOとしているが(X)HTMLへの変換においても同様である。

これらのツールは HTML、RTF、XSL-FO 変換用の XSLT を出力する。このうち FOA[21]は最も早く開発されたオープンソースの XSL-FO 用 XSLT 作成ツールである。特に以下に示すように

```
<xsl:template match="paragraph" foa:class="block">
  <fo:block class="Normal" xsl:use-attribute-sets="Normal">
    <xsl:apply-templates />
  </fo:block>
</xsl:template>
<xsl:attribute-set name="Normal">
```

XSLfast[22]ではサポートしている XSL-FO レンダリングエンジンが FOP 0.20.5、XEP 3.1.X、XSLFormatter 2.5.X・3.X と古いバージョンである点も問題である。ところで XSL ver 1.0 では、1 ページ中に 1 個の本文領域を定義して、そこに 1 つの内容オブジェクトの流し込む（フロー）ことしかできなかった。しかし XSL ver 1.1[4]では、複数の本文領域を定義しておき、それぞれに別のフローを与えることが可能となり、よりデザイン性に富んだレイアウトが可能となった(図 1)。



のようになる。ただし、このような XSLT は第 4 章(5)のような問題がある。

3. データ構造と XSLT の関係

XML をそのデータ構造から分類すると(1)Data-Centric(データ中心型)(2)Document-Centric(ドキュメント中心型)の 2 つのタイプに分類できる[16](表 1)。

表 1 データ中心型・ドキュメント中心型 XML 文書

データ中心型	ドキュメント中心型
データベースにマッピング可能なデータ 住所録等	論文、書籍、カタログ等
<!ELEMENT A (B,C,D)> A の子要素 B C D は B,C,D の順に出現する	<!ELEMENT A (B C D*)> A の子要素 B C D は順不同で何回でも出現する
主にデータ交換、電子商取引	主にオーサリング

データ中心型 XML は(1)要素数・要素の出現順序が確定している、(2)同じ要素が繰り返し出現する構造のため XSLT による要素の抽出・変換は容易である。会員名簿を例にとれば次のようになる。

```
<!ELEMENT meibo (shibu+) >
<!ELEMENT shibu (company+) >
<!ATTLIST shibu name CDATA #REQUIRED>

<!ELEMENT company (name, zip, add, tel, fax,
, url, email, naiyo)>
<!ATTLIST company id CDATA #REQUIRED>

<!ELEMENT name (#PCDATA) >
<!ELEMENT zip (#PCDATA) >
<!ELEMENT add (#PCDATA) >
<!ELEMENT tel (#PCDATA) >
<!ELEMENT fax (#PCDATA) >
<!ELEMENT url (#PCDATA) >
<!ELEMENT email (#PCDATA) >
<!ELEMENT naiyo (#PCDATA) >
```

これはルート要素 meibo の下に複数の支部(shibu 要素)があり、その中に会社(company 要素)が繰り返し存在するという構造である。company 要素以下に社名(name 要素)、郵便番号(zip 要素)、住所(add 要素)等がある。このようなデータの XSLT は次のようになる。

```
<xsl:template match="company">
<xsl:apply-templates select="name"/>
<xsl:apply-templates select="zip"/>
<xsl:apply-templates select="add"/>
<xsl:apply-templates select="tel"/>
<xsl:apply-templates select="fax"/>
<xsl:apply-templates select="url"/>
<xsl:apply-templates select="email"/>
<xsl:apply-templates select="naiyo"/>
</xsl:template>
```

ドキュメント中心型 XML は論文を例にとれば

```
<!ELEMENT article (title, (section|%block;)*
)>
```

```
<!ELEMENT section (title, (subsection|%block;)*)>
```

```
<!ELEMENT subsection (title, (subsubsection|%block;)*)>
```

注) %block; は、リスト、表、パラグラフ、図、写真等のブロック要素である。

のような構造を持っている。つまり、section 子要素として、title 要素が必ず出現し、その次に subsection 要素、ブロック要素が順序に関係なく何回でも出現してもよいという構造である。次にどのような要素が出現するのか明確でないので、その XSLT は次のようになる。

```
<xsl:template match="section">
<xsl:apply-templates />
</xsl:template>
```

この XSLT では<xsl:apply-templates />要素に select 属性がないため、タイトル、リスト、表、パラグラフ、図、写真等のテンプレートが存在すればそれらのテンプレートが実行される。section 要素の子要素が title 要素、table 要素、list 要素である場合、この子要素の順に処理される。つまり XML 木を順にトラバースすることになるので多様なレイアウトの実現は困難である。

4. ドキュメント中心 XML 文書に対する要素レイアウト法

デジタルドキュメントの制作方法論として、構造ベース(structure base)とレイアウトベース(layout base)の 2 つが存在する[17](図 2)。構造ベースとは SGML[7]あるいは XML 文書を対象にした方法であり、前述のスタイルシートを利用してページレイアウト(ヘッダー、フッター、本文エリアなどとして設定した)領域に段落、表、リスト等のオブジェクトを流し込む手法である。さまざまなレイアウトによる出力が可能であるが、大量データの一括処理となり細かなレイアウト設定はできない。

レイアウトベースとは商業印刷分野で行われている DTP(Desk Top Publishing)である。予めレイアウト定義しておき、そのエリアにオブジェクトを流し込むと言う手法をとる。レイアウトは自由自在に設定できるが、手作業であるため生産性のアップは望めない。また文書データの再利用は制限される。

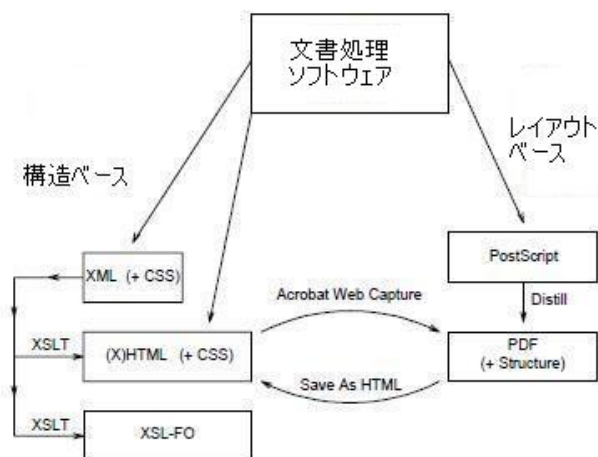


図2 文書処理ソフトウェア[17 を改変]

ドキュメント中心 XML 文書からの組版においても、コスト削減(ページ数削減)・見栄えをよくするためにはある程度のレイアウトベースの要素レイアウト法を取り込む必要がある。例えば、あるセクションを2段組にする例を考えてみる。具体的な方法として以下のような方法が考えられる。(太字部分がレイアウト情報を埋め込んだ部分)

(1) スタイル定義要素を追加する。

```
<div float="left">
  <section>
    -----
  </section>
</div>
<section>
  -----
</section>
```

(2) 要素にスタイル属性を追加する。

```
<section float="left">
  -----
</section>
<section>
  -----
</section>
```

(3) 要素の配置情報を XML の処理命令(Processing Instruction)で対応する。

```
<?float_start type="left"?>
  <section>
    -----
  </section>
<?float_end?>
<section>
  -----
</section>
```

(4) 要素の配置情報を XML のコメントで対応する。

```
<!-- float_start -->
  <section>
    -----
```

```
</section>
<!-- float_end -->
<section>
  -----
</section>
```

が挙げられる。

ここで(1)(2)の場合、「文書構造とレイアウト情報の分離」していないので、One Source, Multi Use という XML の特徴を利用しにくくなってしまいます。例えば、文書中に特定のレイアウト情報が混在することになり、他のレイアウトへの変換が困難となる。(3)(4)の場合、文書構造中にレイアウト情報が混在することは避けられている。それでも XML 文書中に特定の処理情報やコメントを埋め込む必要があり、もし複数のレイアウトで出力しようとする、処理命令やコメントの氾濫によりデータの可読性が著しく損なわれてしまう欠点がある。

これらに対してレイアウト情報を埋め込んだ XSLT を利用すると、XML 文書にはレイアウト情報を一切記述する必要がなくなる。例えば、ある XML 文書を HTML に変換する例を以下に示す。ここで太字部分がレイアウト情報を埋め込んだ部分である。

```
<xsl:template match="section">
  <xsl:if test="position()=1">
    <xsl:text disable-escaping="yes">&lt;div s
    tyle="float: left;" &gt;</xsl:text>
  </xsl:if>
  <xsl:if test="position()=2">
    <xsl:text disable-escaping="yes">&lt;/div&
    gt;</xsl:text>
  </xsl:if>
  <xsl:apply-templates />
</xsl:template>
```

この例から推察できるように、レイアウトが複雑になればなるほど XSLT 文書は複雑になり、作成の負担は増大する。またレイアウトを変更のたびに XSLT 文書を作成する必要があるため、これらのメンテナンスが煩雑である。そこで XML 文書の再利用、つまり One Source, Multi Use と同様に、XSLT 文書自身が再利用可能になれば、上記の問題は軽減されると期待される。

5. 要素のマッピング

5.1. 再利用可能な XSLT

XML 文書を様々なレイアウトに変換するには、それぞれのレイアウトに対応した XSLT 文書を作成しなければならない(図3)。

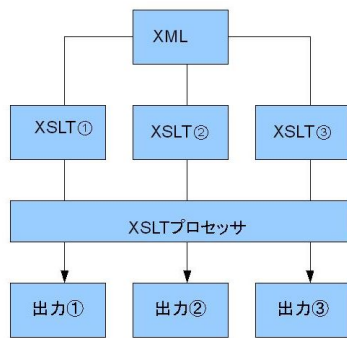


図3 従来の変換手法

Rich なレイアウトを実現するためには各 XSLT 文書に配置の情報と書式情報を記述しなければならないが、前章で触れたように、これでは「再利用困難な XSLT 文書」を作ることとなる。

さて一般的なドキュメント中心 XML 文書では、part, chapter, section 等のタグにより文書の基本的な構造を、さらにそれらの子要素としてパラグラフ、リスト、図、表等のタグで文書の基本構造単位を記述している。Rich なレイアウトとは、これら基本構造単位の配置と書式を定義することである。そこで、XSLT 文書の内容は、文書の基本構造単位をどのように表示するか、に関する書式にとどめておき、配置についての情報は別のファイルとして分離する。これにより、XSLT 文書は Rich なレイアウトを実現するための再利用可能な部品として利用できるようになる。一方、XSLT 文書から除かれた配置に関する情報はマッピングファイルと呼ばれるファイルに記述される。マッピングファイルには基本構造単位の配置情報と、実際に基本構造単位を表示する際に利用する XSL 文書の対応付けを記述する。マッピングファイルは変換ツールを利用して XSLT 文書に変換され、さらに定義済みの基本構造単位用の XSLT 文書と統合される。これを XML 文書に適用して出力する（図 4）。これが本研究の中心的な技法である。つまりレイアウトの変更はマッピングファイルの変更だけで可能となる。XML 文書に対してレイアウト情報の要素を付加する必要もなく、また書式については再利用可能な XSLT 部品を利用することができるので、Rich なレイアウトを容易に実現できる。

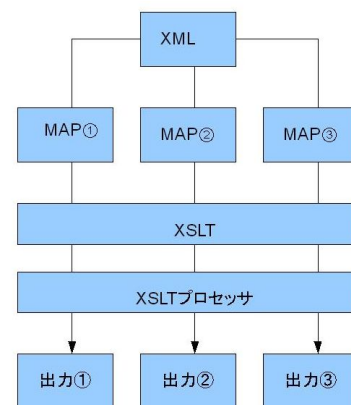


図4 変換にマッピングファイルを介在させる

今回、マッピングファイルの機能としては変換対象 XML 部分木の（１）表形式や段組への再構成、（２）フォントサイズ等の書式設定の変更を実装した（第 6 章参照）。

またこのことにより次のような問題を解決することができる。たとえば、ある要素の直下に存在する 1 番目の段落と 2 番目の段落のフォントサイズを変更する場合、`<xsl:template match="A/P[1]">`、`<xsl:template match="A/P[2]">`のような 2 個のテンプレートを作成することになるがこのような XSLT は特定のノード集合のためのテンプレートであるため再利用ができない。このことを回避するには書式設定属性をパラメータとしてマッピングファイルに記載すればよい（`<Topic class="p" href="/A/P[1]" font-size="10pt" />`、`<Topic class="p" href="/A/P[2]" font-size="15pt" />`）。段落テンプレートではこれらパラメータを受取り、書式設定する。このことはパラグラフ、リスト、図、表等の基本構造単位をライブラリー化することに他ならない。

5.2. マッピングファイル変換用 XSLT の概要

例として SECT2 以下の 1 番目と 2 番目の SECT4 要素を 2 段組に変換する XSLT を想定してみる(図 5)。

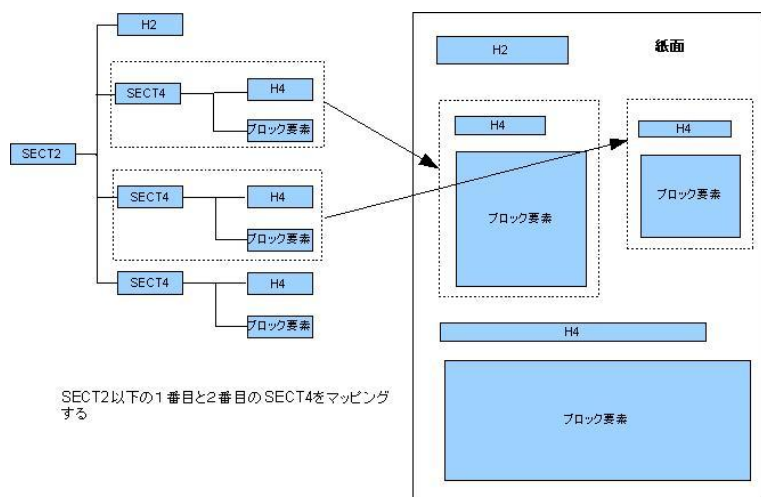


図5 要素のマッピング

```
<Topic class="dan" column-count="2" column-gap="10mm"> ①
  <Topic class="h4" href="/SECT2/SECT4[1]/H4" />②
  <Topic class="その他の要素に対応したクラス" href="その他の要素を選択する XPath 式" />③
  <Topic class="h4" href="/SECT2/SECT4[2]/H4" />④
  <Topic class="その他の要素に対応したクラス" href="その他の要素を選択する XPath 式" />⑤
</Topic>
```

図6 マッピングファイル

まず2段組に変換する要素グループを dan クラスでラッピングする。ここで段数、段間幅をセットする(図6①)。次に変換対象のノードを Topic 要素の href 属性に Xpath 式で、対応する表示クラスを class 属性に設定する(図6②～⑤)。

```
<!-- class="dan"
  <Topic class="dan" column-count="2" column-gap="6mm" >
    <Topic class="dan" column-count="3" column-gap="6mm" >
  -->
<xsl:template match="Topic[@class='dan']">
<xsl:variable name="column-count" select="@column-count" /> ①
<xsl:variable name="column-gap" select="@column-gap" /> ②
<fo:block-container axf:column-count="{ $column-count}" axf:column-gap="{ $column-gap}">③
  <xsl:apply-templates />④
</fo:block-container>
</xsl:template>
```

図7 "dan"テンプレート

図6①で設定された段数、段間幅を"dan"テンプレート側でパラメータとして受け取る(図7①、②)。こうすることで段数、段間幅の変更が発生してもマッピング

グファイルの変更で対応できる。このパラメータを XSL-FO の fo:block-container に設定する(図7③)。図7④の<xsl:apply-templates />により"h4"テンプレート、"その他の要素に対応した"テンプレート(略)をマッピングファイルに記載された順にコールする。"dan"テンプレートでの XSLT 変換プロセスをまとめると次のようになる。

- (1) xsl:apply-templates により次に処理する子ノードのリストを得る。
- (2) 子ノードのリストに対応するテンプレートを見つけ、テンプレートから結果木の断片を作成する。
- (3) (2)に戻り子ノードのリストがなくなるまで処理を続ける。

```
<!-- class="h4"
  <Topic class="h4" href="/SECT2/SECT4[1]/H4" />
-->
<xsl:template match="Topic[@class='h4']">
<xsl:variable name="href_name" select="@href" />
<xsl:variable name="exp" select="concat('document(',$ap,$doc_file,$ap,')',$href_name)" />①
<xsl:apply-templates select="saxon:evaluate($exp)"> ②
  <xsl:with-param name="arg_span" select="string(@span)" />③
  <xsl:with-param name="arg_font-stretch" select="string(@font-stretch)" /> ④
  <xsl:with-param name="arg_font-size" select="string(@font-size)" /> ⑤
  <xsl:with-param name="arg_line-height" select="string(@line-height)" /> ⑥
  <xsl:with-param name="arg_break-type" select="string(@break-type)" /> ⑦
</xsl:apply-templates>
</xsl:template>
```

図8 "h4"テンプレート

図8①で実際の変換対象 XML が特定される。この場合、href 属性が/SECT2/SECT4[1]/H4 なので実際のノードは document('変換対象 XML')/SECT2/SECT4[1]/H4 となる。この部分が変数となっている理由は、クラスが h4 として書式設定するのは常に H4 要素であるとは限らないからである。③～⑦でパラメータを設定して②で"H4"テンプレートをコールする。

```
<xsl:template match="H4">
<xsl:param name="arg_span" />
<xsl:param name="arg_font-stretch" />
<xsl:param name="arg_font-size" />
<xsl:param name="arg_line-height" />
<xsl:param name="arg_break-type" />
```

```

<fo:block xsl:use-attribute-sets="H4" > ①
  <xsl:call-template name="block_att_set">
    <xsl:with-param name="arg_span" select="$arg_span"/>
    <xsl:with-param name="arg_font-stretch" select="$arg_font-stretch"/>
    <xsl:with-param name="arg_font-size" select="$arg_font-size"/>
    <xsl:with-param name="arg_line-height" select="$arg_line-height"/>
    <xsl:with-param name="arg_break-type" select="$arg_break-type"/>
  </xsl:call-template>
  <xsl:apply-templates />
</fo:block>
</xsl:template>

```

図 9 "H4"テンプレート

デフォルトの組版体裁情報は、図 9①で設定してあるが初期設定を変更するために図 8③～⑦でフォントサイズ、行間等を変更可能なようにこれらをパラメータとしてマッピングファイルから受け取る。

テンプレート間の属性パラメータの受け渡しをまとめると次のようになる（表 2）。

表 2 テンプレート間の属性パラメータの受け渡し

テンプレート	INPUT 属性	出力 XSL-FO 要素	コールされるテンプレート	コールされるテンプレートに渡される属性
dan	column-count, column-gap	fo:block-container	h4, その他の要素に対応するテンプレート (略)	href, その他の書式設定
h4	href, その他の書式設定	----	H4	その他の書式設定
H4	その他の書式設定	fo:block	italic, boldなどのインライン	----

6. マッピングファイルの概要

マッピングファイルは Map 要素をルートに、Topic 要素と処理コントロール要素を子要素としてもつ XML 形式のデータであり、変換対象 XML のどの部分を選択しどのように書式設定するかを記述したものである（付録 C、D 参照）。

マッピングファイルは TopicIF などの要素コントロールグループと要素表示・再構成グループで構成されている。後者は class 属性により機能と設定可能な属性を識別している。例えば、<Topic class="block" width="178mm" height="3mm"/>は幅 178mm 高さ 3mm の空白領域を確保するクラスであり width 属性 height 属性とも必須である。また、<Topic class="h6" href="/SECT

2/SECT4[1]/SECT6[1]/H6"/>は href 属性が必須であるが、font-stretch 属性、font-size 属性、line-height 属性、break-type 属性は任意であり、break-type 属性値は page、column のどちらかである。

6.1. 要素表示グループ

class 属性、href 属性は必須である。href 属性に記述した XPath 式でマッピングを行う。class 属性値によりパラグラフ、見出し、リスト、図、表等を識別し、設定可能なパラメータは class 属性値に依存する（表 3 参照）。

表 3 主要素表示グループ

class 属性	機能	設定属性	必須・任意	内容
h3 h4 h5 h6 h7	タイトルの表示	href	必須	ロケーションパス式
		font-stretch	任意	文字の縮小値(単位%)
		font-size	任意	文字サイズ(単位 mm)
		line-height	任意	行高(単位 mm)
		break-type	任意	break-type="page"(改ページ) break-type="column"(改段)
例	SECT2 の下部要素である最初の SECT4 の下にある H4 を h4(見出しレベル 4)にマッピングする。表示スタイルはデフォルト。 <Topic class="h4" href="/SECT2/SECT4[1]/H4" /> フォントサイズを 15mm に変更する場合、<Topic class="h4" href="/SECT2/SECT4[1]/H4" font-size="15mm"/>			
main_p	P(本文)要素の表示	href	必須	ロケーションパス式
		type	任意	type="list" であればリスト形式で表示。l_number (ラベル文字数) 属性が必要
		font-stretch	任意	文字の縮小値(単位%)
		font-size	任意	文字サイズ(単位 mm)
		line-height	任意	行高(単位 mm)
例	<Topic class="main_p" href="/SECT2/SECT4[1]/SECT5[1]/P"/>			
att_list_caution	P 要素を注釈として表示	href	必須	ロケーションパス式
		att_list_str	必須	リスト項目とする文字列
		font-size	任意	文字サイズ(単位 mm)
		line-height	任意	行高(単位 mm)
例	注)文字列文字列文字列文字列文字列文字列文字列文字列 文字列文字列文字列文字列文字列文字列文字列文字列 文字列 <Topic class="att_list_caution" att_list_str="注)" href="/SECT2/SECT4[3]/SECT5[1]/P"/>			

rbw-table	RBW-TABLE(表)の表示	href	必須	ロケーションパス式
		aki_before	任意	セル中のコンテンツと上部セル境界とのアキ (単位 mm)
		aki_after	任意	セル中のコンテンツと下部セル境界とのアキ (単位 mm)
		width	任意	表の表示幅 デフォルトでは版面幅の表を創る (単位 mm)
		frame	任意	表の枠線表示 デフォルトでは枠線表示 frame="none" 枠線非表示 frame="sides" 左右の枠線のみ表示
		font-size	任意	表の文字サイズ (単位 mm)
		line-height	任意	表の行の高さ (単位 mm)
		col_color_pos	任意	カラー表示する列の番号 (col_color_pos="1 2 3")
		omitheader	任意	改ページ後 HEADER を非表示にする default は表示 omitheader="yes"
例	<Topic class="rbw-table" href="SECT2/SECT4[4]/SECT5[1]/RBW-TABLE" aki_before="0.25mm" aki_after="0.25mm" col_color_pos="1 2" font-size="2.55mm" />			
fig	FIG(画像)に含まれる全ての要素を表示	href	必須	ロケーションパス式
		width	任意	画像の横幅 (単位 mm, %)
		height	任意	画像の縦幅 (単位 mm, %)
		fig.align	任意	画像部分のみ left, center, right
		fig.t.align	任意	画像のタイトル部分のみ left, center, right
例	<Topic class="fig" href="SECT2/FiG[2]" />			

6.2. 要素再構成グループ

class 属性は必須である。href 属性は存在しない。class="dan"、class="table" の 2 つである (表 4 参照)。

表 4 要素再構成グループ

class 属性	機能	設定属性	必須・任意	内容
dan	段組の設定	column-count	必須	段数 (数字)
		column-gap	必須	段間距離 (単位 mm)
例	SECT2 の下部要素である 3、4 番目の SECT4 を 2 段組にしその次に横罫線をひく <Topic class="dan" column-count="2" column-gap="10mm"> <Topic class="h4" href="/SECT2/SECT4[3]/H4"/> <Topic class="fig" href="/SECT2/SECT4[3]/FIG[1]" width="84mm" height="60mm"/> <Topic class="h4" href="/SECT2/SECT4[4]/H4"/> <Topic class="fig" href="/SECT2/SECT4[4]/FIG[1]" width="84mm" height="60mm"/> </Topic>			

<Topic class="hr" />				
table	XML コンテンツを表に再構成する	frame	任意	表の枠線の表示 (デフォルトでは枠線を表示する。frame="none" で枠線を非表示に)
		width	任意	表の横幅 (単位 mm) width があれば固定幅の表を表示。デフォルトでは版面幅の表を創る
		例	<Topic class="table" frame="none" width="90mm">	
colspec	表のカラム幅の設定	colnum	必須	表のカラム番号 (数字)
		width	必須	表のカラム幅 (単位 mm)
例	<Topic class="colspec" colnum="1" width="32mm" />			
tr	行の高さ設定	height	任意	行の高さ (単位 mm)
例	<Topic class="tr" height="35mm">			
td	表のセルの設定	align	任意	セル (水平方向) の表示 left, center, right
		valign	任意	セル (垂直方向) の表示 top, middle, bottom
例	<Topic class="td" valign="top" align="center"> 表の例 <Topic class="table" width="60mm"> <Topic class="colspec" colnum="1" width="55mm" /> <Topic class="tr"> <Topic class="td" valign="middle" align="center"> <Topic class="main_p" href="SECT2/P[1]" /> </Topic> </Topic>			

6.3. 処理コントロールグループ

繰り返し処理のための Topic_For_each 要素、マクロ展開のための Topic_Call 要素がある。条件付き処理のために Topic_Choose・Topic_When・Topic_Otherwise・Topic_IF 要素がある (表 5 参照)。

表 5 処理コントロールグループ

要素名	機能	属性	必須・任意	内容
Topic_For_each	繰り返し処理を行う。Topic_For_each の入れ子可能	for	必須	index 変数 先頭に \$ がつく
		from	必須	数字
		to	必須	数字
例	SECT2 要素の直下の SECT4 要素の下にある 1 番目の SECT5 要素から 3 番目の SECT5 要素の下 H5 要素を 2 桁の数字付き箇条書に変換する <Topic_For_each for="\$i" from="1" to="3"> <Topic class="h5" href="/SECT2/SECT4[1]/SECT5[\$i]/H5" type="list" l_number="2"/> </Topic_For_each>			
Topic_IF	Topic_For_each の中で条件判定を行う	test	必須	XPath 算術式 例では index 変数が 1 または 2 の時、改段する
例	<Topic_For_each for="\$i" from="1" to="6"> <Topic class="h5" href="/SECT2/SECT4[10]/SECT5[\$i]/H5" type="list" l_number="2"/>			

	<Topic class="main_p" href="/SECT2/SECT4[10]/SECT5[\$i]/P" />> <TopicIF test="(\$i=1) or (\$i=2)"> <Topic class="break" type="column" /> </TopicIF> </Topic_For_each>		
Topic_Chose	条件選択判定を行う		子要素として、Topic_When, Topic_Otherwise を持つ
例	<Topic_For_each for="\$i" from="1" to="11"> <Topic_Chose> <Topic_When test="\$i=10 or \$i=11"> <Topic class="block" width="65mm" height="1mm" /> <Topic class="table" width="65mm"> <Topic class="colspec" colnum="1" width="65mm" /> <Topic class="tr"> <Topic class="td" valign="middle"> <Topic class="main_p" href="/SECT2/SECT4[1]/P[\$i]" /> </Topic> </Topic> </Topic_When> <Topic_Otherwise> <Topic class="main_p" href="/SECT2/SECT4[1]/P[\$i]" /> </Topic_Otherwise> </Topic_Chose> </Topic_For_each>		
Topic_When	条件分岐を行う。子要素として、Topic を持つ	test	必須 XPath 算術式
Topic_Otherwise	どのTopic_Whenにも該当しない場合の処理を行う		子要素として、Topic を持つ
Topic_Call	文字列の置換を行う。子要素として、Topic_With_Param を持つ	href	必須 展開するマッピングファイルのファイルパス
Topic_With_Param	name を val で置換する	name	必須 文字列
		val	必須 文字列
例	Topic_For_each の中では使用できない。 子要素として、Topic_With_Param がない場合。href のデータを単純にコピーする。 子要素として、Topic_With_Param を持つ場合。 zzz.sub の #1 -----> 4 zzz.sub の #2 -----> 5 zzz.sub の #3 -----> 6 zzz.sub の s4_pos -----> 7 に置換する <Topic_Call href="zzz.sub"> <Topic_With_Param name="#1" val="4" /> <Topic_With_Param name="#2" val="5" /> <Topic_With_Param name="#3" val="6" />		

<Topic_With_Param name="s4_pos" val="7" /> </Topic_Call>

6.4. その他のグループ

強制改ページ・改段(class="break")、空白領域の確保(class="block")、水平線(class="hr")の表示、作成済み XS L-FO ファイルのインクルード(class="include")などがある (表 6 参照)。

表 6 その他のグループ

class 属性	機能	設定属性	必須・任意	内容
hr	水平線を設定する	width	任意	水平線の横幅 (単位 mm)
		aki_before	任意	水平線の前の空き幅 (単位 mm)
		aki_after	任意	水平線の後の空き幅 (単位 mm)
例	<Topic class="hr" /> <Topic class="hr" aki_before="1mm" width="50mm" /> デフォルト値は spce-before=2mm, spce-after=2mm, leader-length=100%			
break	改頁または改段する	type	必須	type="page" (改ページ) type="column" (改段)
例	<Topic class="break" type="page" /> <Topic class="break" type="column" />			
block	ブロック (空白領域) を設定する	width	必須	ブロックの横幅 (単位 mm)
		height	必須	ブロックの高さ (単位 mm)
例	<Topic class="block" width="178mm" height="3mm" />			
include	作成済みの FO ファイルをインクルードする	ref	必須	インクルードするファイルのパス。ファイル形式は FO ファイル。<fo:block> を TOP 要素とする
例	<Topic class="include" ref="sub_fo/include.fo" />			

7. マッピングファイルの検証

7.1. XML の検証

XML 文書構造の定義には様々なスキーマが利用されている。DTD、W3C XML Schema、Relax NG などがあるが、これらは XML 文書の取り得る構造を記述したものである。つまり、要素の出現順序、出現回数、親子関係、属性は必須か、そうでないか等を定義している。W3C XML Schema、Relax NG は要素内容・属性値について制約を定義 (文字列型、整数型、日付型等) できるので、要素内容に対する厳密な検証が可能となった。しかし次のような欠点がある[26]。

- (1) 属性の選択を指定すること。
たとえば、A 要素は B 属性または C 属性のいずれかを持つ必要がある。
- (2) 要素と属性をモデルグループにグループ化すること。

xs:sequence、xs:choice、xs:all などを使用して要素をグループ化することは可能だが、要素と属性の両方をグループ化することはできない。たとえば、要素と属性のセットの選択を作成することはできない。

- (3) 要素または属性の値に基づいてコンテンツモデルを変化させること。
たとえば、status 属性の値が available の場合、要素は uptime 子要素を持つ必要がある。そうでない場合は、downtime 子要素を持つ必要がある。

次のような（表 7）ルールは既存のスキーマ言語では検証困難である。

表 7 ルールベースの検証が必要な XML

例 1	制約ルール
<pre><?xml version="1.0"?> <Person Title="Mr"> <Name>Eddie</Name> <Gender>Male</Gender> </Person></pre>	(1) Person 要素は必須属性 Title をもつ。 (2) 子要素として Name, Gender 要素をもつ。 (3) 子要素 Name の次に Gender 要素が出現しなければならない。 (4) Title 属性値が Mr ならば Gender 要素は Male でなければならない。
例 2	制約ルール
<pre><?xml version="1.0"?> <Persentage> <Result name="John">61</Result> <Result name="Sara">24</Result> <Result name="Bill">15</Result> </Persentage></pre>	(1) Result 要素のトータルが 100 でなければならない。

7.2. スキマトロンとは

マッピングファイルのように class 属性によって設定可能な属性が変化する場合、DTD、W3C Schema などの文法ベースの DSDL(Document Schema Definition Languages : 文書スキーマ定義言語)では検証不可能でありルールベースの DSDL で検証する必要がある。

スキマトロンは文法の代わりに XPath 表現を使用して XML 文書中に使用できる要素・属性を定義するルールベースの言語である。スキマトロンは XML の文法を作成する代わりに文書内の特定コンテキストに適用される表明(assertion)を作成する。表明が偽の場合、スキーマ作成者が提供する診断メッセージが表示される(図 10)。

```
<?xml version="1.0" encoding="utf-8"?>
<schema xmlns="http://www.ascc.net/xml/schematron">
```

```
<pattern name="パターン名 (optional)">
  <rule context="XPath 式">
    <assert test="XPath 式">非成立時に表示される文</assert>
    <report test="XPath 式">成立時に表示される文</report>
    ...
  </rule>
  ...
</pattern>
...
</schema>
```

図 10 スキマトロンの基本構造

今回はスキマトロン (Schematron) 処理系[24]でマッピングファイルを検証した。スキマトロン処理系は「スキーマ定義 XML を入力として XSLT を出力する XSLT」処理系であるため 2 段階のプロセスが必要である[25](図 11)。

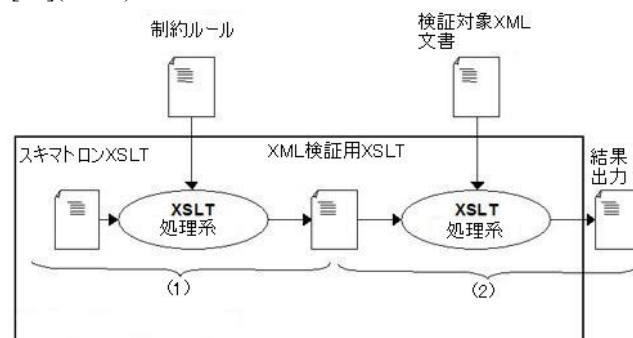


図 11 スキマトロン処理系[25]を改変

- (1) 制約ルールをスキマトロンスキーマに記載しスキマトロン処理系(XSLT)により XML 検証用 XSLT を作成する。

例えば、

```
<Topic class="block" width="178mm" height="3mm"/>
```

の制約ルールは次のようになる。

```
<!-- Topic[@class='block'] must width, height -->
<rule context="Topic[@class='block']"> ①
  <assert test="count(@*)=3"> ②
    must contains 3 attributes (include class attribute) ! ③
    <assert test="(count(@width)=1) and (count(@height)=1)"> ④
      must contains width attribute and height attribute ⑤
    </assert>
  </assert>
</rule>
```

このルールの意味は以下の通りである。rule 要素の context 属性に検証対象要素を Xpath 式で記述する(①)。この場合、class 属性値が block である Topic 要素を意

味する。要素や属性のチェックは `assert` 要素の `test` 属性で行う。`test` 属性には XPath 式、XSLT 組み込み関数を使用する。②で属性の個数を、④で `width` 属性、`height` 属性が必須であることをチェックしている。

(2) 次に XML 検証用 XSLT で検証対象 XML を検証する。

属性数が3個でない場合③が、`width` 属性、`height` 属性の両方がない場合⑤のエラーメッセージが出力される。

8. 処理概要

本章ではマッピングファイルを利用した Rich レイアウトの実現法を示す。

8.1. XSLT、XSL-FO 処理系について

要素のマッピング情報は、Topic 要素の `href` 属性から、`<Topic class="h4" href="/SECT2/SECT4[1]/H4"/>` のように参照し XSLT のテンプレートでは `href` 属性値を変数として受け取る。XSLT1.0 の仕様では、`xsl:apply-templates` 要素の `select` 属性は実行時に確定しているノード集合でなければならないので、`select` 属性を可変な文字列として受け取れない。このような仕様を実装している処理系として Xalan[11]、Saxon[18]などがある。今回は豊富な拡張関数を実装している Saxon を使用した。`saxon:evaluate` 関数は、XPath 式を現在の XPath 式コンテキストで評価した結果を返す。

XSL-FO 処理系としてオープンソースの FOP[19]、商用製品として XEP Engine[20]、XSL Formatter[10]があるが、FOP は XSL-FO ver 1.0 及び 1.1 の仕様を完全に実装しておらず、実用的な組版には問題がある。今回、日本語組版の独特なルール（行頭・行末禁則処理等）をサポートしている XSL Formatter ver 4.2 を使用した。

8.2. 処理方法

STEP 1 : 処理対象の XML 文書からマップファイルの骨格 (tenkai.xml) を自動的に生成する。

target.xml (処理対象の XML)

```
<SECT2>
-----
<FIG>.....</FIG>
<SECT4>
  <H4>...</H4>
  <P>...</P>
</SECT4>
-----
</SECT2>
```

```
java -jar c:\saxonb9\saxon9.jar -s:target.xml
  -xsl:make_path.xsl -o:tenkai.xml (付録C参照)
```

tenkai.xml

```
<Map>
-----
  <Topic class="fig" href="/SECT2/FIG[1]"/>
  <Topic class="h4" href="/SECT2/SECT4[1]/H4[1]"/>
  <Topic class="main_p" href="/SECT2/SECT4[1]/P[1]"/>
-----
</Map>
```

STEP 2 : マップファイルの編集。

一連の要素を多段組に設定、画像サイズの変更等を行う。

・編集前（上下に配置された体系図）

```
<!-- ***** 体系図 1***** -->
<Topic class="h4" href="/SECT2/SECT4[3]/H4[1]"/>
<Topic class="fig" href="/SECT2/SECT4[3]/FIG[1]"/>
<!-- ***** 体系図 2***** -->
<Topic class="h4" href="/SECT2/SECT4[4]/H4[1]"/>
<Topic class="fig" href="/SECT2/SECT4[4]/FIG[1]"/>
```

・編集後（体系図を2段組にし図サイズを変更）

```
<!-- ***** 体系図 1,2 ***** -->
<Topic class="dan" column-count="2" column-gap="10mm">
  <Topic class="h4" href="/SECT2/SECT4[3]/H4" />
  <Topic class="fig" href="/SECT2/SECT4[3]/FIG[1]" width="84mm" height="60mm" />
  <Topic class="h4" href="/SECT2/SECT4[4]/H4" />
  <Topic class="fig" href="/SECT2/SECT4[4]/FIG[1]" width="84mm" height="60mm" />
</Topic>
```

STEP 3 : マップファイルのスキマトロン処理系による検証。

(1)スキマトロンスキーマ(tenkai.sch 付録 J 参照)から検証用 XSLT(validate_tenkai.xsl)を作成する。

```
java -jar c:\saxonb9\saxon9.jar -s:tenkai.sch
  -xsl:c:\schematron1-5\schematron-basic.xsl
  -o:validate_tenkai.xsl
```

(2)マップファイル(tenkai.xml)を検証用 XSLT(validate_tenkai.xsl)で検証する。

```
java -jar c:\saxonb9\saxon9.jar -s:tenkai.xml
  -xsl:validate_tenkai.xsl
```

STEP 4 : マップファイル (tenkai.xml) 中の ENTITY, Topic_For_each, TopicIF 等を展開し tenkai.dat を出力(図 12)。

tenkai.xml (展開前)

```
<Topic_For_each for="$i" from="1" to="3">
<Topic class="h5" href="/SECT2/SECT4[1]/SECT
5[$i]/H5" type="list" l_number="2"/>
</Topic_For_each>
```

```
java -jar c:\saxonb9\saxon9.jar -s:tenkai.xml
l -xsl:tenkai.xsl -o:tenkai.dat
```

tenkai.dat (展開後)

```
<Topic class="h5" href="/SECT2/SECT4[1]/SECT
5[1]/H5" type="list" l_number="2"/>
<Topic class="h5" href="/SECT2/SECT4[1]/SECT
5[2]/H5" type="list" l_number="2"/>
<Topic class="h5" href="/SECT2/SECT4[1]/SECT
5[3]/H5" type="list" l_number="2"/>
```

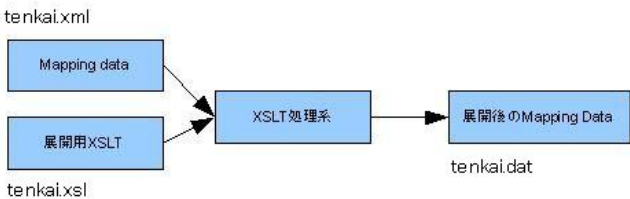


図 12 マッピングデータの展開

STEP 5 : tenkai.dat を map_jpn_fo.xml で xsl-fo データに変換する(図 13)。

パラメーター doc_file : 処理対象となる xml データ

```
java -jar c:\saxonb9\saxon9.jar -s:tenkai.dat
-xsl:map_jpn_fo.xsl -o:output.fo doc_file=
処理対象となる xml データの file path
```

STEP 6 : tenkai.fo を XSL Formatter で PDF に出力する (図 13)。

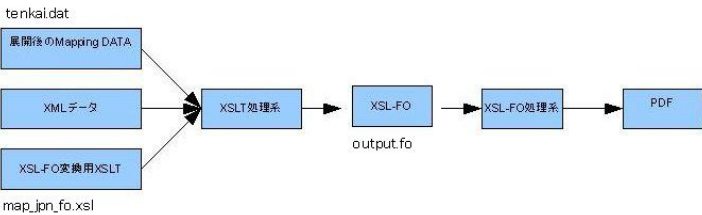


図 13 XSL-FO の PDF 変換

8.3. テストケース・付録

テストケースとして単純なカタログを例とした (付録 A、付録 B 参照)。付録 E は test_map.xml をそのまま PDF 変換したもので見栄えが良くない。付録 F は test_map.xml に要素レイアウトを設定し (付録 D 参照) PDF 変換したものである。特長と用途を 3 段組にし、体系

図 1 と体系図 2 を 2 段組にしている。こうすることでコンパクトで見やすいカタログに変換できた。表 8 と図 14 に付録リストと XML、XSLT の関連図を付記する。

表 8 付録リスト

付録 A	catalog.dtd (カタログ文書型定義)
付録 B	test.xml (テスト用 XML 文書)
付録 C	test_map.xml (make_path.xsl よりの出力)
付録 D	test_map1.xml (test_map.xml をレイアウト編集したもの)
付録 E	test_map.xml より作成された PDF
付録 F	test_map1.xml より作成された PDF
付録 G	make_path.xsl (XML 文書からマッピングファイル生成用 XSLT)
付録 H	tenkai.xsl (処理コントロールグループを含むマッピングファイルを正規化する XSLT)
付録 I	make_jpn_fo.xsl (マッピングファイルから XSL-FO 生成用 XSLT)
付録 J	tenkai.sch (マッピングファイル検証用スキマトロンスキーマ)

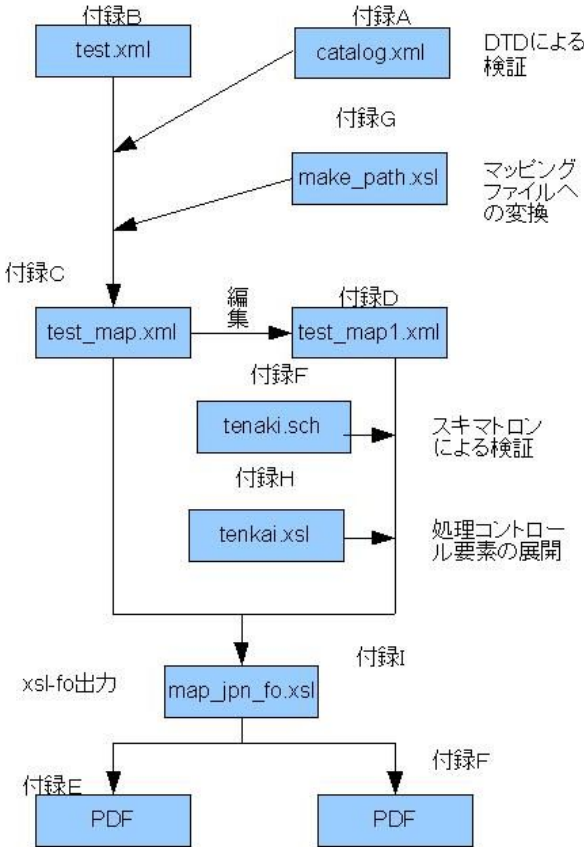


図 14 付録収載 XML、XSLT の関連図

9. 考察・今後の課題

本プロトタイプでは catalog.dtd (付録 A 参照) に準拠した大量の XML 文書の変換を目指したものである。XML 文書に対して複数のプログラマーが XSLT を作

成した場合、生産性のアップは望めないし作成された XSLT の品質のばらつきが大きい。またレイアウト変更が発生した場合のデバッグが煩雑である。マッピングファイルに変換順序、文書構成ルールを記述することでプログラム経験のない人員でも容易に PDF を作成できる利点は非常に大きい。しかしマッピングファイルの作成・編集には XSLT、XPath の知識を前提としているため GUI 環境を持ったツールが必要である。今後の課題として GUI 環境の実装に取り組む予定である。

次に様々な DTD に準拠した XML 文書への対応であるが、本プロトタイプのように XSLT テンプレートが固定したものではなく可変でなければならない。まず考えられるのは変換対象要素に対応した XSLT テンプレートをなんらかの方法で生成することである。例えば、DOC/CHAPTER[1]/HEADER に位置する HEADER 要素に対しては、

```
<xsl:template match="DOC/CHAPTER[1]/HEADER">
  <fo:block xsl:use-attribute-sets="header">
    <xsl:apply-templates/>
  </fo:block>
</xsl:template>
```

のような XSLT テンプレートを作成し、各要素の XSLT テンプレートを 1 つの XSLT に纏め上げればよい。ただし、名前付き属性集合は予め作成しなければならない。次に、例として 3 番目と 4 番目の SECT4 要素を 2 段組することを考えてみる。まず、XML 文書と次のような XSLT テンプレートデータを用意する。

XML 文書

```
----- 略 -----
<SECT4>
<H4>体系図 1</H4>
<FIG SRC="rpcc.gif"/>
</SECT4>
<SECT4>
<H4>体系図 2</H4>
<FIG SRC="rppn.gif"/>
</SECT4>
----- 略 -----
```

XSLT テンプレートデータ

```
<?xml version="1.0" encoding="Shift_JIS"?>
<make_xsl>
-- 略 --
<setxsl class="mulcolumn_st" column-count="2"
  " column-gap="10mm" path="/SECT2/SECT4[3]"/>
①
<setxsl class="mulcolumn_en" path="/SECT2/SECT4[5]"/>②
<setxsl class="H4" path="//H4"/>③
<setxsl class="FIG" path="//FIG"/>④
```

```
-- 略 --
</make_xsl>
```

①と④で fo:block-container の開始、終了タグを挿入し、②、④で path 属性に対応した XSLT テンプレートと class 属性に対応した名前付き属性集合を設定する。これらのデータより下記のような XSLT テンプレートが生成できればよい。

XSLT テンプレート

```
<xsl:template match="/SECT2">
<fo:root>
  <xsl:apply-templates />
</fo:root>
</xsl:template>

<xsl:template match="/SECT2/SECT4[3]">
<xsl:text disable-output-escaping="yes">
  &lt;fo:block-container axf:column-count="2"
  axf:column-gap="10mm"&gt;
</xsl:text>
  <xsl:apply-templates />
</xsl:template>

<xsl:template match="/SECT2/SECT4[5]">
<xsl:text disable-output-escaping="yes">
  &lt;/fo:block-container&gt;
</xsl:text>
  <xsl:apply-templates />
</xsl:template>

<xsl:template match="//H4">
<fo:block xsl:use-attribute-sets="H4">
  <xsl:value-of select="." />
  <xsl:apply-templates />
</fo:block>
</xsl:template>

<xsl:template match="//FIG">
<fo:block xsl:use-attribute-sets="FIG">
  <fo:external-graphic src="{@SRC}">
  </fo:external-graphic>
</fo:block>
</xsl:template>
```

つまり XSLT テンプレートデータから XSLT テンプレートを作成する XSLT を用意すればよい。また、直接、GUI 環境を持ったプログラムから XSLT テンプレートと名前付き属性集合を生成できれば初心者に易しいであろう。用紙サイズの変更、表紙・偶数・奇数ページの設定、各ページのヘッダー、フッター領域等の設定が必要である。今後の課題としたい。

謝 辞

本研究にあたりご指導いただきました大阪市立大学大学院創造都市研究科の中野 秀男教授、北 克一教

授、Venkatesh RAGHAVAN 教授、大西 克実准教授に感謝申し上げます。

また、本研究を進めるにあたって適切な助言および貴重なご意見をいただきました京都大学地域研究総合情報センター 原正一郎教授に深く御礼申し上げます。

参考文献・サイト

- [1] Extensible Markup Language(XML) : <http://www.w3.org/TR/REC-xml/>
- [2] W3C Cascading Style Sheets home page : <http://www.w3.org/Style/CSS/>
- [3] XSL Transformations (XSLT) Version 1.0 : <http://www.w3.org/TR/xslt>
- [4] Extensible Stylesheet Language (XSL) Version 1.1 : <http://www.w3.org/TR/xsl/>
- [5] Adobe PostScript language specifications : http://partners.adobe.com/public/developer/ps/index_specs.html
- [6] LaTeX (A document preparation system) : <http://www.latex-project.org/>
- [7] Standard Generalized Markup Language (SGML) : <http://www1.u-netsurf.ne.jp/~711rl/SGMLindex.html>
- [8] RTF Word 2007: Rich Text Format (RTF) Specification, version 1.9.1 : <http://www.microsoft.com/downloads/details.aspx?familyid=DD422B8D-FF06-4207-B476-6B5396A18A2B&displaylang=en>
- [9] XML Path Language (XPath) Version 1.0 : <http://www.w3.org/TR/xpath>
- [10] Antenna House XSL Formatter V4.2 : <http://www.antenna.co.jp/XSL-FO/>
- [11] Xalan-C++ version 1.10 : <http://xml.apache.org/xalan-c/>
- [12] Xalan-C++ extension library : <http://xml.apache.org/xalan-c/extensionslib.html>
- [13] CSS Paged Media Level 3 : <http://www.w3.org/TR/css3-page/>
- [14] Antenna House Formatter V5.0: <http://www.antenna.co.jp/AHF/index.html>
- [15] Prince : <http://www.princexml.com/>
- [16] Bourret, R. : XML and Databases. <http://www.rpbouret.com/xml/XMLAndDatabases.htm>
- [17] Matthew R. B. Hardy, David F. Brailsford : Mapping and displaying structural transformations between XML and PDF. Proceedings of the 2002 ACM symposium on Document engineering : 95-102, 2002.
- [18] Saxon-B 9.1 <http://saxon.sourceforge.net/>
- [19] FOP : <http://xmlgraphics.apache.org/fop/>
- [20] XEP Engine : <http://www.renderx.com/>
- [21] F. Giannetti : FOA: an XSL-FO Authoring Tool, Hewlett-Packard Laboratories - HP LABORATORIES TECHNICAL REPORT HPL, 2001
- [22] XSLfast : <http://www.xslfast.com/>
- [23] Altova StyleVision2008 : http://www.altova.com/products/stylevision/xslt_stylesheet_designer.html
- [24] Schematron 1.5 : <http://www.ascc.net/xml/resource/schematron/>
- [25] Robertsson, E. : An Introduction to Schematron <http://www.xml.com/lpt/a/1318>
- [26] Dare Obasanjo : Schematron を使用して XML ドキュメント検証を向上する : <http://msdn.microsoft.com/ja-jp/library/aa468554.aspx>

付録A catalog.dtd

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<!-- カタログ用 DTD -->
<!ENTITY % text "(#PCDATA|B|I|U|SUB|SUP|RBW-ANCHOR|RBW-XREF)*">
<!ENTITY % block "((P|P.L|FIG|RBW-TABLE)+)">
<!ENTITY % graphic "((PHOTO|PICT)+)">

<!-- <SECT2> 個別商品タグ (ルート) -->
<!ELEMENT SECT2 (H2,(SECT3|SECT4|SECT5|(%block;))*>
<!ELEMENT SECT3 (H3,(SECT4|SECT5|SECT6|(%block;))*>
<!ELEMENT SECT4 (H4,(SECT5|SECT6|(%block;))*>
<!ELEMENT SECT5 (H5,(SECT6|(%block;))*>
<!ELEMENT SECT6 (H6,((%block;))*>
<!ELEMENT H2 %text;>
<!ELEMENT H3 %text;>
<!ELEMENT H4 %text;>
<!ELEMENT H5 %text;>
<!ELEMENT H6 %text;>
<!ELEMENT P %text;>
<!-- 文書の冒頭に出現するPR文章用のタグ。-->
<!ELEMENT P.L %text;>
<!-- 図項目 -->
<!ELEMENT FIG
(FIG.T, (%block;)*, FIG.NAME, (%graphic;)?, (%block;))*>
<!ELEMENT FIG.T %text;>
<!ELEMENT FIG.NAME (#PCDATA)>
<!ELEMENT PHOTO (#PCDATA)>
<!ELEMENT PICT (#PCDATA)>
<!--
FIG 画像表示に関わるタグを一括りにする親タグ。
FIG.T 画像のタイトル
FIG.NAME 画像のファイル名
PHOTO 写真
PICT 線画
-->
<!-- 表項目 -->
<!ELEMENT RBW-TABLE (TGROUP+)>
<!ATTLIST RBW-TABLE FRAME (TOP|BOTTOM|TOPBOT|ALL|SIDES|NONE) #IMPLIED
LEFT-INDENT CDATA #IMPLIED
ID ID #IMPLIED
COLSEP NMTOKEN #IMPLIED
ROWSEP NMTOKEN #IMPLIED>
<!-- 表の中で最上位層のタグ。-->

<!ELEMENT TGROUP (COLSPEC*, SPANSPEC*, THEAD?, TFOOT?, TBODY)>
<!ATTLIST TGROUP COLS NMTOKEN #REQUIRED
COLSEP NMTOKEN #IMPLIED
ROWSEP NMTOKEN #IMPLIED>
<!-- 表で、列数分の<COLSPEC>といくつかの<TBODY>を
内包するタグ。自身はCOLS属性で列数情報を持つ。
-->
<!ELEMENT COLSPEC EMPTY>
```

```
<!ATTLIST COLSPEC COLNUM NMTOKEN #IMPLIED
COLNAME NMTOKEN #IMPLIED
ALIGN (LEFT|RIGHT|CENTER|JUSTIFY|CHAR) #IMPLIED
VALIGN (TOP|MIDDLE|BOTTOM) #IMPLIED
COLWIDTH CDATA #IMPLIED
COLSEP NMTOKEN #IMPLIED>
<!-- 表で、親タグの<TGROUP>で指定された列数分くりかえし出現し、属性でそれぞれの列の名前、列の幅等の情報を持つ。中身はなし。
-->
<!ELEMENT SPANSPEC EMPTY>
<!ATTLIST SPANSPEC NAMEST NMTOKEN #IMPLIED
NAMEEND NMTOKEN #IMPLIED
SPANNAME NMTOKEN #REQUIRED>
<!ELEMENT THEAD (COLSPEC*, RBW-ROW+)>
<!ELEMENT TFOOT (COLSPEC*, RBW-ROW+)>
<!ELEMENT TBODY (RBW-ROW+)>
<!-- 表で、いくつかのRBW-ROWを内包するタグ。-->
<!ELEMENT RBW-ROW (ENTRY)+>
<!ATTLIST RBW-ROW ROWSEP NMTOKEN #IMPLIED>
<!-- 表の1行ごとに開始、終了するタグ。-->
<!ELEMENT ENTRY (%block;)+>
<!ATTLIST ENTRY COLNAME NMTOKEN #IMPLIED
MOREROWS NMTOKEN '0'
VALIGN (TOP|MIDDLE|BOTTOM) #IMPLIED
ALIGN (LEFT|RIGHT|CENTER|JUSTIFY|CHAR) #IMPLIED
NAMEST NMTOKEN #IMPLIED
NAMEEND NMTOKEN #IMPLIED>
<!-- 表の1セルごとに開始、終了するタグ。-->
<!-- ボールド、イタリック、下線、下付き、上付き -->
<!ELEMENT B %text;>
<!ELEMENT I %text;>
<!ELEMENT U %text;>
<!ELEMENT SUB %text;>
<!ELEMENT SUP %text;>
<!-- ハイパーリンク項目 -->
<!ELEMENT RBW-XREF %text;>
<!ATTLIST RBW-XREF REFID IDREF #REQUIRED
TYPE CDATA #IMPLIED>
<!-- リンク元に使用するタグ。リンク先IDの情報を
入れるREFID属性を持つ。-->
<!ELEMENT RBW-ANCHOR EMPTY>
<!ATTLIST RBW-ANCHOR ID ID #REQUIRED>
<!-- リンク先となる場所に使用。中身はなく、ID属性を持つ。-->
```

付録B test.xml

```
<?xml version="1.0" encoding="Shift_JIS"?>
<!DOCTYPE SECT2 SYSTEM "catalog.dtd">
<SECT2>
<H2><RBW-ANCHOR ID="test"></RBW-ANCHOR>商品名称</H2>
<P.L>商品説明。商品説明。</P.L>
<P.L>商品説明。商品説明。商品説明。商品説明。商品説明。</P.L>
<FIG>
<FIG.T></FIG.T>
<FIG.NAME>rp1c.gif</FIG.NAME>
<PHOTO>##</PHOTO>
</FIG>
<SECT4>
<H4>特長</H4>
<SECT5>
<H5>1.特長 1 特長 1 特長 1 特長 1 特長 1。</H5>
<P> 特長 1 のテキスト特長 1 のテキスト特長 1 のテキスト
特長 1 のテキスト特長 1 のテキスト特長 1 のテキスト。</P>
</SECT5>
<SECT5>
<H5>2.特長 2 特長 2 特長 2 特長 2。</H5>
<P> 特長 2 のテキスト特長 2 のテキスト特長 2 のテキスト
特長 2 のテキスト。</P>
</SECT5>
<SECT5>
<H5>3.特長 3 特長 3。</H5>
<P> 特長 3 のテキスト特長 3 のテキスト特長 3 のテキスト
特長 3 のテキスト特長 3 のテキスト特長 3 のテキスト特長
3 のテキスト特長 3 のテキスト。</P>
</SECT5>
<SECT5>
<H5>4.特長 4 特長 4 特長 4 特長 4 特長 4 特長 4 特長 4。</H5>
</SECT5>
<SECT5>
<H5>5.特長 5 特長 5 特長 5。</H5>
</SECT5>
</SECT4>
<SECT4>
<H4>用途</H4>
<SECT5>
<H5>1.用途 1 用途 1 用途 1 用途 1。</H5>
</SECT5>
<SECT5>
<H5>2.用途 2 用途 2 用途 2 用途 2。</H5>
</SECT5>
<SECT5>
<H5>3.用途 3 用途 3 用途 3 用途 3。</H5>
</SECT5>
</SECT4>
<SECT4>
<H4>体系図 1</H4>
<FIG>
<FIG.T></FIG.T>
<FIG.NAME>rpcc.gif</FIG.NAME>
<PICT>##</PICT>
</FIG>
</SECT4>
```

```
<SECT4>
<H4>体系図 2</H4>
<FIG>
<FIG.T></FIG.T>
<FIG.NAME>rppn.gif</FIG.NAME>
<PICT>##</PICT>
</FIG>
</SECT4>
<SECT4>
<H4>品種</H4>
<P>この表の説明この表の説明この表の説明この表の説明</P>
<RBW-TABLE LEFT-INDENT="-1">
<TGROUP COLS="6">
<COLSPEC COLNAME="1" COLWIDTH="61p" ALIGN="C
ENTER" VALIGN="MIDDLE"></COLSPEC>
<COLSPEC COLNAME="2" COLWIDTH="82p" ALIGN="C
ENTER" VALIGN="MIDDLE"></COLSPEC>
<COLSPEC COLNAME="3" COLWIDTH="63p" ALIGN="C
ENTER" VALIGN="MIDDLE"></COLSPEC>
<COLSPEC COLNAME="4" COLWIDTH="63p" ALIGN="C
ENTER" VALIGN="MIDDLE"></COLSPEC>
<COLSPEC COLNAME="5" COLWIDTH="63p" ALIGN="C
ENTER" VALIGN="MIDDLE"></COLSPEC>
<COLSPEC COLNAME="6" COLWIDTH="63p" ALIGN="C
ENTER" VALIGN="MIDDLE"></COLSPEC>
<THEAD>
<RBW-ROW>
<ENTRY COLNAME="1" MOREROWS="1"><P>カラム 1</P></ENTRY>
<ENTRY COLNAME="2" MOREROWS="1"><P>カラム 2</P></ENTRY>
<ENTRY NAMEST="3" NAMEEND="4" MOREROWS="0"><P>カラム 3－4カラム 3－4</P><P>カラム 3－4カラム 3－4</P></ENTRY>
<ENTRY NAMEST="5" NAMEEND="6" MOREROWS="0"><P>カラム 5－6</P><P>カラム 5－6カラム 5－6</P></ENTRY>
</RBW-ROW>
<RBW-ROW>
<ENTRY COLNAME="3" MOREROWS="0"><P>カラム 3</P></ENTRY>
<ENTRY COLNAME="4" MOREROWS="0"><P>カラム 4</P></ENTRY>
<ENTRY COLNAME="5" MOREROWS="0"><P>カラム 5</P></ENTRY>
<ENTRY COLNAME="6" MOREROWS="0"><P>カラム 6</P></ENTRY>
</RBW-ROW>
</THEAD>
<TBODY>
<RBW-ROW>
<ENTRY COLNAME="1" MOREROWS="7"><P>カラム 1</P></ENTRY>
<ENTRY COLNAME="2" MOREROWS="0"><P>カラム 2</P></ENTRY>
<ENTRY COLNAME="3" MOREROWS="0"><P>カラム 3</P></ENTRY>
<ENTRY COLNAME="4" MOREROWS="0"><P>カラム 4</P></ENTRY>
```

```

<ENTRY COLNAME="5" MOREROWS="0"><P>カラム 5</
P></ENTRY>
<ENTRY COLNAME="6" MOREROWS="0"><P>カラム 6</
P></ENTRY>
</RBW-ROW>
<RBW-ROW>
<ENTRY COLNAME="2" MOREROWS="0"><P>カラム 2</
P></ENTRY>
<ENTRY COLNAME="3" MOREROWS="0"><P>カラム 3</
P></ENTRY>
<ENTRY COLNAME="4" MOREROWS="0"><P>カラム 4</
P></ENTRY>
<ENTRY COLNAME="5" MOREROWS="0"><P>カラム 5</
P></ENTRY>
<ENTRY COLNAME="6" MOREROWS="0"><P>カラム 6</
P></ENTRY>
</RBW-ROW>

```

略

```

<RBW-ROW>
<ENTRY COLNAME="2" MOREROWS="0"><P>カラム 2</
P></ENTRY>
<ENTRY COLNAME="3" MOREROWS="0"><P>カラム 3</
P></ENTRY>
<ENTRY COLNAME="4" MOREROWS="0"><P>カラム 4</
P></ENTRY>
<ENTRY COLNAME="5" MOREROWS="0"><P>カラム 5</
P></ENTRY>
<ENTRY COLNAME="6" MOREROWS="0"><P>カラム 6</
P></ENTRY>
</RBW-ROW>
</TBODY>
</TGROUP>
</RBW-TABLE>
</SECT4>
<SECT4>
<H4>定格</H4>
<SECT5>
<H5>1.○○○○仕様</H5>
<RBW-TABLE LEFT-INDENT="-1">
<TGROUP COLS="7">
<COLSPEC COLNAME="1" COLWIDTH="69p" ALIGN="C
ENTER" VALIGN="MIDDLE"></COLSPEC>
<COLSPEC COLNAME="2" COLWIDTH="71p" ALIGN="C
ENTER" VALIGN="MIDDLE"></COLSPEC>
<COLSPEC COLNAME="3" COLWIDTH="71p" ALIGN="C
ENTER" VALIGN="MIDDLE"></COLSPEC>
<COLSPEC COLNAME="4" COLWIDTH="71p" ALIGN="C
ENTER" VALIGN="MIDDLE"></COLSPEC>
<COLSPEC COLNAME="5" COLWIDTH="71p" ALIGN="C
ENTER" VALIGN="MIDDLE"></COLSPEC>
<COLSPEC COLNAME="6" COLWIDTH="71p" ALIGN="C
ENTER" VALIGN="MIDDLE"></COLSPEC>
<COLSPEC COLNAME="7" COLWIDTH="71p" ALIGN="C
ENTER" VALIGN="MIDDLE"></COLSPEC>
<THEAD>
<RBW-ROW>
<ENTRY COLNAME="1" MOREROWS="0"><P>カラム 1</
P></ENTRY>
<ENTRY COLNAME="2" MOREROWS="0"><P>カラム 2</

```

```

P></ENTRY>
<ENTRY COLNAME="3" MOREROWS="0"><P>カラム 3</
P></ENTRY>
<ENTRY COLNAME="4" MOREROWS="0"><P>カラム 4</
P></ENTRY>
<ENTRY COLNAME="5" MOREROWS="0"><P>カラム 5</
P></ENTRY>
<ENTRY COLNAME="6" MOREROWS="0"><P>カラム 6</
P></ENTRY>
<ENTRY COLNAME="7" MOREROWS="0"><P>カラム 7</
P></ENTRY>
</RBW-ROW>
</THEAD>
<TBODY>
<RBW-ROW>
<ENTRY COLNAME="1" MOREROWS="0"><P>カラム 1</
P></ENTRY>
<ENTRY COLNAME="2" MOREROWS="7"><P>カラム 2</
P></ENTRY>
<ENTRY COLNAME="3" MOREROWS="7"><P>カラム 3</
P></ENTRY>
<ENTRY COLNAME="4" MOREROWS="0"><P>カラム 4</
P></ENTRY>
<ENTRY COLNAME="5" MOREROWS="0"><P>カラム 5</
P></ENTRY>
<ENTRY COLNAME="6" MOREROWS="0"><P>カラム 6</
P></ENTRY>
<ENTRY COLNAME="7" MOREROWS="6"><P>カラム 7</
P></ENTRY>
</RBW-ROW>

```

略

```

<RBW-ROW>
<ENTRY COLNAME="1" MOREROWS="0"><P>カラム 1</
P></ENTRY>
<ENTRY COLNAME="4" MOREROWS="0"><P>カラム 4</
P></ENTRY>
<ENTRY COLNAME="5" MOREROWS="0"><P>カラム 5</
P></ENTRY>
<ENTRY COLNAME="6" MOREROWS="0"><P>カラム 6</
P></ENTRY>
<ENTRY COLNAME="7" MOREROWS="0"><P>カラム 7</
P></ENTRY>
</RBW-ROW>
</TBODY>
</TGROUP>
</RBW-TABLE>
<P> ※表の注釈表の注釈表の注釈</P>
</SECT5>
</SECT4>
</SECT2>

```

付録C test_map.xml

```
<?xml version="1.0" encoding="Shift_JIS"?>
<Map id="test">
  <Topic class="TopTitle" pl_font-size="4mm"
    pl_line-height="5mm"/>
  <Topic class="block" width="178mm" height="
    3mm"/>
  <Topic class="fig" href="/SECT2/FIG[1]"/>
  <!-- ***** 特長 ***** -->
  <Topic class="h4" href="/SECT2/SECT4[1]/H
    4[1]"/>
  <Topic class="h5" href="/SECT2/SECT4[1]/S
    ECT5[1]/H5[1]"/>
  <Topic class="main_p" href="/SECT2/SECT4[
    1]/SECT5[1]/P[1]"/>
  <Topic class="h5" href="/SECT2/SECT4[1]/S
    ECT5[2]/H5[1]"/>
  <Topic class="main_p" href="/SECT2/SECT4[
    1]/SECT5[2]/P[1]"/>
  <Topic class="h5" href="/SECT2/SECT4[1]/S
    ECT5[3]/H5[1]"/>
  <Topic class="main_p" href="/SECT2/SECT4[
    1]/SECT5[3]/P[1]"/>
  <Topic class="h5" href="/SECT2/SECT4[1]/S
    ECT5[4]/H5[1]"/>
  <Topic class="h5" href="/SECT2/SECT4[1]/S
    ECT5[5]/H5[1]"/>
  <!-- ***** 用途 ***** -->
  <Topic class="h4" href="/SECT2/SECT4[2]/H
    4[1]"/>
  <Topic class="h5" href="/SECT2/SECT4[2]/S
    ECT5[1]/H5[1]"/>
  <Topic class="h5" href="/SECT2/SECT4[2]/S
    ECT5[2]/H5[1]"/>
  <Topic class="h5" href="/SECT2/SECT4[2]/S
    ECT5[3]/H5[1]"/>
  <!-- ***** 体系図 1 ***** -->
  <Topic class="h4" href="/SECT2/SECT4[3]/H
    4[1]"/>
  <Topic class="fig" href="/SECT2/SECT4[3]/
    FIG[1]"/>
  <!-- ***** 体系図 2 ***** -->
  <Topic class="h4" href="/SECT2/SECT4[4]/H4[
    1]"/>
  <Topic class="fig" href="/SECT2/SECT4[4]/
    FIG[1]"/>
  <!-- ***** 品種 ***** -->
  <Topic class="h4" href="/SECT2/SECT4[5]/H
    4[1]"/>
  <Topic class="main_p" href="/SECT2/SECT4[
    5]/P[1]"/>
  <Topic class="rbw-table" href="/SECT2/SEC
    T4[5]/RBW-TABLE[1]" aki_before="0.25mm"
    aki_after="0.25mm"
    col_color_pos="1"/>
  <!-- ***** 定格 ***** -->
  <Topic class="h4" href="/SECT2/SECT4[6]/H
    4[1]"/>
  <Topic class="h5" href="/SECT2/SECT4[6]/S
    ECT5[1]/H5[1]"/>
  <Topic class="rbw-table" href="/SECT2/SEC
```

```
T4[6]/SECT5[1]/RBW-TABLE[1]"
    aki_before="0.25mm"
    aki_after="0.25mm"
    col_color_pos="1"/>
  <Topic class="main_p" href="/SECT2/SECT4[
    6]/SECT5[1]/P[1]"/>
</Map>
```

付録D test_map1.xml

```
<?xml version="1.0" encoding="Shift_JIS"?>
<!DOCTYPE Map [
<!ENTITY MaxWidth "178mm" >
<!ENTITY Column-Gap-For3 "6.5mm" >
<!ENTITY Column-Width-For3 "55mm" >
]>
<Map id="test">
  <Topic class="TopTitle" pl_font-size="4mm"
    pl_line-height="5mm" />
  <Topic class="block" width="&MaxWidth;" height="3mm" />
  <!-- ***** 特長 ***** -->
  <Topic class="dan" column-count="3" column-gap="&Column-Gap-For3;">
    <Topic class="fig" href="/SECT2/FIG[1]"
      fig.align="center" />
    <Topic class="block" width="&Column-Width-For3;" height="2mm" />
    <Topic class="include" href="ref/rohs.fo" />
    <Topic class="h4" href="/SECT2/SECT4[1]/H4"/>
    <Topic_For_each for="$i" from="1" to="5">
      <Topic class="h5" href="/SECT2/SECT4[1]/SECT5[$i]/H5" type="list" l_number="2"/>
      <Topic class="main_p" href="/SECT2/SECT4[1]/SECT5[$i]/P" />
    </Topic_For_each>
    <Topic class="block" width="&Column-Width-For3;" height="2mm" />
  <!-- ***** 用途 ***** -->
  <Topic class="h4" href="/SECT2/SECT4[2]/H4"/>
  <Topic_For_each for="$i" from="1" to="3">
    <Topic class="h5" href="/SECT2/SECT4[2]/SECT5[$i]/H5" type="list" l_number="2"/>
  </Topic_For_each>
</Topic>
<Topic class="hr" />
  <!-- ***** 体系1 ***** -->
  <Topic class="dan" column-count="2" column-gap="10mm">
    <Topic class="h4" href="/SECT2/SECT4[3]/H4" />
    <Topic class="fig" href="/SECT2/SECT4[3]/FIG[1]" width="84mm" height="60mm" />
    <Topic class="h4" href="/SECT2/SECT4[4]/H4" />
    <Topic class="fig" href="/SECT2/SECT4[4]/FIG[1]" width="84mm" height="60mm" />
  </Topic>
  <Topic class="hr" />
  <!-- ***** 体系2 ***** -->
  <Topic_Call href="sub/hinsyu_hako.sub">
    <Topic_With_Param name="#width1" val="89mm" />
    <Topic_With_Param name="#width2" val="89mm" />
    <Topic_With_Param name="s4_pos" val="5" />
  </Topic_Call>
</Map>
```

```
</Topic_Call>
  <Topic class="rbw-table" href="/SECT2/SECT4[5]/RBW-TABLE" aki_before="0.25mm" aki_after="0.25mm" col_color_pos="1" />
  <Topic class="hr" />
  <!-- ***** 定格 ***** -->
  <Topic class="h4" href="/SECT2/SECT4[6]/H4" />
  <Topic class="h5" href="/SECT2/SECT4[6]/SECT5[1]/H5" />
  <Topic class="rbw-table" href="/SECT2/SECT4[6]/SECT5[1]/RBW-TABLE" aki_before="0.25mm" aki_after="0.25mm" col_color_pos="1" />
  <Topic class="att_p" href="/SECT2/SECT4[6]/SECT5[1]/P" />
</Map>
```


付録G make_path.xsl

```
<?xml version="1.0" encoding="Shift_JIS"?>
<xsl:stylesheet version="1.0"
xmlns:saxon="http://saxon.sf.net/"
xmlns:fn="http://www.w3.org/2005/xpath-functions"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
exclude-result-prefixes="saxon fn">

<xsl:output method="xml" indent="yes" encoding="Shift_JIS" />

<xsl:template match="SECT2">
<xsl:variable name="s2_id" select="fn:lower-case(H2/RBW-ANCHOR/@ID)" />
<Map id="{s2_id}">
  <Topic class="TopTitle" pl_font-size="4mm"
  " pl_line-height="5mm"/>
  <Topic class="block" width="178mm" height="3mm"/>
  <xsl:apply-templates />
</Map>
</xsl:template>

<xsl:template match="H4">
<xsl:comment>
  ***** <xsl:value-of select="." /> **
  *****
</xsl:comment>
<xsl:variable name="class_name" select="fn:lower-case(local-name())" />
<xsl:variable name="path" select="saxon:path()" />
<Topic class="{class_name}" href="{path}" />
</xsl:template>

<xsl:template match="H5|H6|H7|H8">
<xsl:variable name="class_name" select="fn:lower-case(local-name())" />
<xsl:variable name="path" select="saxon:path()" />
<Topic class="{class_name}" href="{path}" />
</xsl:template>

<xsl:template match="RBW-TABLE">
<xsl:variable name="class_name" select="fn:lower-case(local-name())" />
<xsl:variable name="path" select="saxon:path()" />
<Topic class="{class_name}" href="{path}"
aki_before="0.25mm" aki_after="0.25mm" color_pos="1" />
</xsl:template>

<xsl:template match="FIG">
<xsl:variable name="class_name" select="fn:lower-case(local-name())" />
```

```
<xsl:variable name="path" select="saxon:path()" />
<Topic class="{class_name}" href="{path}" />
</xsl:template>

<xsl:template match="P">
<xsl:variable name="class_name" select="'main_p'" />
<xsl:variable name="path" select="saxon:path()" />
<Topic class="{class_name}" href="{path}" />
</xsl:template>

<xsl:template match="text()" />
</xsl:stylesheet>
```

付録H tenkai.xsl

```
<?xml version="1.0" encoding="Shift_JIS"?>
<xsl:stylesheet version="1.0"
xmlns:saxon="http://saxon.sf.net/"
xmlns:fn="http://www.w3.org/2005/xpath-functions"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:common="http://exslt.org/common"
exclude-result-prefixes="saxon fn common">

<xsl:output method="xml" indent="yes" encoding="Shift_JIS" />

<!-- *****
MAP ファイル中の Topic_For_each, TopicIF
Topic_Choose, Topic_When, Topic_Otherwise
を展開する
*****
* -->
<xsl:template match="Map">
<xsl:copy>
<xsl:copy-of select="@*" />
<xsl:apply-templates />
</xsl:copy>
</xsl:template>

<xsl:template match="Topic">
<xsl:copy>
<xsl:copy-of select="@*" />
<xsl:apply-templates />
</xsl:copy>
</xsl:template>

<xsl:template match="comment()">
<xsl:copy />
</xsl:template>
<!-- ***** -->
<!-- Topic_For_each -->
<!-- ***** -->
<xsl:template match="Topic_For_each">
<xsl:variable name="index" select="@for" />
<xsl:variable name="from" select="@from" />
<xsl:variable name="to" select="@to" />

<xsl:variable name="DD">
<xsl:call-template name="Topic_For_each_Treatment">
<xsl:with-param name="arg_index" select="$index" />
<xsl:with-param name="arg_from" select="$from" />
<xsl:with-param name="arg_to" select="$to" />
</xsl:call-template>
</xsl:variable>

<xsl:variable name="cnt_Topic_For_each" select="count(common:node-set($DD)/descendant-or-self::Topic_For_each)" />
```

```
<xsl:choose>
<xsl:when test="$cnt_Topic_For_each > 0">
<xsl:apply-templates select="common:node-set($DD)/*" />
</xsl:when>
<xsl:otherwise>
<xsl:copy-of select="$DD" />
</xsl:otherwise>
</xsl:choose>
</xsl:template>

<xsl:template name="Topic_For_each_Treatment">
<xsl:param name="arg_index" />
<xsl:param name="arg_from" />
<xsl:param name="arg_to" />

<xsl:if test="$arg_from <= $arg_to">
<xsl:apply-templates select="*" mode="in_loop">
<xsl:with-param name="arg_index" select="$arg_index" />
<xsl:with-param name="arg_from" select="$arg_from" />
</xsl:apply-templates>

<xsl:call-template name="Topic_For_each_Treatment">
<xsl:with-param name="arg_index" select="$arg_index" />
<xsl:with-param name="arg_from" select="$arg_from + 1" />
<xsl:with-param name="arg_to" select="$arg_to" />
</xsl:call-template>
</xsl:if>
</xsl:template>

<xsl:template match="Topic_For_each" mode="in_loop">
<xsl:param name="arg_index" />
<xsl:param name="arg_from" />

<xsl:variable name="from_att" select="normalize-space(@from)" />

<xsl:variable name="from_tmp">
<xsl:call-template name="replace-string">
<xsl:with-param name="text" select="$from_att" />
<xsl:with-param name="replace" select="$arg_index" />
<xsl:with-param name="with" select="$arg_from" />
</xsl:call-template>
</xsl:variable>

<xsl:variable name="to_att" select="normalize-space(@to)" />
```

```

<xsl:variable name="to_tmp">
  <xsl:call-template name="replace-string">
    <xsl:with-param name="text" select="$to_a
tt" />
    <xsl:with-param name="replace" select="$a
rg_index" />
    <xsl:with-param name="with" select="$arg_
from" />
  </xsl:call-template>
</xsl:variable>

<xsl:copy>
  <xsl:copy-of select="@*" />
  <xsl:attribute name="from"><xsl:value-of se
lect="saxon:evaluate($from_tmp)" /></xsl:att
ribute>
  <xsl:attribute name="to"><xsl:value-of sele
ct="saxon:evaluate($to_tmp)" /></xsl:attribu
te>

  <xsl:apply-templates select="*" mode="in_l
oop">
    <xsl:with-param name="arg_index" sel
ect="$arg_index" />
    <xsl:with-param name="arg_from" sel
ect="$arg_from" />
  </xsl:apply-templates>
</xsl:copy>
</xsl:template>

<xsl:template match="Topic" mode="in_loop">
<xsl:param name="arg_index" />
<xsl:param name="arg_from" />

<xsl:copy>
  <xsl:copy-of select="@*" />
  <xsl:if test="contains(@href,$arg_index)">
    <xsl:variable name="tml">
      <xsl:call-template name="replace-strin
g">
        <xsl:with-param name="text" select="
@href" />
        <xsl:with-param name="replace" selec
t="$arg_index" />
        <xsl:with-param name="with" select="
$arg_from" />
      </xsl:call-template>
    </xsl:variable>
    <xsl:attribute name="href"><xsl:value-o
f select="$tml" /></xsl:attribute>
  </xsl:if>

  <xsl:apply-templates select="*" mode="in_l
oop">
    <xsl:with-param name="arg_index" sel
ect="$arg_index" />
    <xsl:with-param name="arg_from" sel
ect="$arg_from" />
  </xsl:apply-templates>
</xsl:copy>
</xsl:template>

```

```

<xsl:template match="TopicIF" mode="in_loop"
>
<xsl:param name="arg_index" />
<xsl:param name="arg_from" />

<xsl:variable name="test_att" select="normal
ize-space(@test)" />

<xsl:variable name="tmp">
  <xsl:call-template name="replace-string">
    <xsl:with-param name="text" select="$test
_att" />
    <xsl:with-param name="replace" select="$a
rg_index" />
    <xsl:with-param name="with" select="$arg_
from" />
  </xsl:call-template>
</xsl:variable>

<xsl:choose>
  <xsl:when test="contains($tmp, '$')">
    <xsl:copy>
      <xsl:attribute name="test"><xsl:value-o
f select="$tmp" /></xsl:attribute>
      <xsl:apply-templates select="*" mode="in
_loop">
        <xsl:with-param name="arg_index" selec
t="$arg_index" />
        <xsl:with-param name="arg_from" selec
t="$arg_from" />
      </xsl:apply-templates>
    </xsl:when>
    <xsl:otherwise>
      <xsl:if test="saxon:evaluate($tmp)">
        <xsl:apply-templates select="*" mode="
in_loop">
          <xsl:with-param name="arg_index" sel
ect="$arg_index" />
          <xsl:with-param name="arg_from" sel
ect="$arg_from" />
        </xsl:apply-templates>
      <!--      <xsl:copy-of select="child::*" /
>      -->
    </xsl:if>
  </xsl:otherwise>
</xsl:choose>
</xsl:template>

<xsl:template match="Topic_Choose" mode="in_
loop">
<xsl:param name="arg_index" />
<xsl:param name="arg_from" />

<xsl:apply-templates select="Topic_When" mod
e="in_loop">
  <xsl:with-param name="arg_index" select="$a
rg_index" />
  <xsl:with-param name="arg_from" select="$a

```

```

rg_from" />
</xsl:apply-templates>

<xsl:if test="Topic_Otherwise">
  <xsl:variable name="test_data">
    <xsl:for-each select="Topic_When">
      <xsl:variable name="test_att" select="normalize-space(@test)" />
      <xsl:value-of select="concat('(', $test_att, ')')" />
    </xsl:for-each>
  </xsl:variable>

  <xsl:variable name="tmp">
    <xsl:call-template name="replace-string">
      <xsl:with-param name="text" select="$test_data" />
      <xsl:with-param name="replace" select="$arg_index" />
      <xsl:with-param name="with" select="$arg_from" />
    </xsl:call-template>
  </xsl:variable>

  <xsl:if test="not(saxon:evaluate($tmp))">
    <xsl:apply-templates select="Topic_Otherwise" mode="in_loop">
      <xsl:with-param name="arg_index" select="$arg_index" />
      <xsl:with-param name="arg_from" select="$arg_from" />
    </xsl:apply-templates>
  </xsl:if>
</xsl:if>
</xsl:template>

<xsl:template match="Topic_When" mode="in_loop">
  <xsl:param name="arg_index" />
  <xsl:param name="arg_from" />

  <xsl:variable name="test_att" select="normalize-space(@test)" />

  <xsl:variable name="tmp">
    <xsl:call-template name="replace-string">
      <xsl:with-param name="text" select="$test_att" />
      <xsl:with-param name="replace" select="$arg_index" />
      <xsl:with-param name="with" select="$arg_from" />
    </xsl:call-template>
  </xsl:variable>

  <xsl:choose>
    <xsl:when test="contains($tmp, '$')">
      <xsl:copy>
      <xsl:attribute name="test"><xsl:value-of select="$tmp" /></xsl:attribute>

```

```

    <xsl:apply-templates select="*" mode="in_loop">
      <xsl:with-param name="arg_index" select="$arg_index" />
      <xsl:with-param name="arg_from" select="$arg_from" />
    </xsl:apply-templates>
  </xsl:when>
  <xsl:otherwise>
    <xsl:if test="saxon:evaluate($tmp)">
      <xsl:apply-templates select="*" mode="in_loop">
        <xsl:with-param name="arg_index" select="$arg_index" />
        <xsl:with-param name="arg_from" select="$arg_from" />
      </xsl:apply-templates>
    <!-- <xsl:copy-of select="child::*" /> -->
  </xsl:if>
</xsl:otherwise>
</xsl:choose>
</xsl:template>

<xsl:template match="Topic_Otherwise" mode="in_loop">
  <xsl:param name="arg_index" />
  <xsl:param name="arg_from" />

  <xsl:apply-templates select="*" mode="in_loop">
    <xsl:with-param name="arg_index" select="$arg_index" />
    <xsl:with-param name="arg_from" select="$arg_from" />
  </xsl:apply-templates>
</xsl:template>

<!-- ***** -->
<!-- Topic_Call -->
<!-- ***** -->

<xsl:template match="Topic_Call">
  <xsl:variable name="href" select="string(@href)" />

  <xsl:variable name="subfile" select="document($href)" />
  <xsl:variable name="Topic_With_Param_cnt" select="//Topic_With_Param" />

  <xsl:choose>
    <xsl:when test="$Topic_With_Param_cnt = 0">
      <xsl:copy-of select="$subfile/Topic_Sub/*" />
    </xsl:when>
    <xsl:otherwise>
      <xsl:variable name="params">
        <xsl:for-each select="Topic_With_Param">

```

```

        <para>
          <xsl:copy-of select="@*" />
        </para>
      </xsl:for-each>
    </xsl:variable>

    <xsl:apply-templates select="$subfile/Topic_Sub/*" mode="subfile">
      <xsl:with-param name="arg_params" select="common:node-set($params)/*" />
    </xsl:apply-templates>
  </xsl:otherwise>
</xsl:choose>
</xsl:template>

<!-- ***** -->
<!-- Topic mode=subfile -->
<!-- ***** -->
<xsl:template match="Topic" mode="subfile">
  <xsl:param name="arg_params" />

  <xsl:copy>
    <xsl:copy-of select="@*" />
    <xsl:for-each select="@*">
      <xsl:variable name="att_name" select="name()" />
      <xsl:variable name="att_val" select="." />

      <xsl:variable name="result">
        <xsl:call-template name="convert">
          <xsl:with-param name="arg_params" select="$arg_params" />
          <xsl:with-param name="arg_att_val" select="$att_val" />
        </xsl:call-template>
      </xsl:variable>

      <xsl:attribute name="{ $att_name }"><xsl:value-of select="$result" /></xsl:attribute>

    </xsl:for-each>

    <xsl:apply-templates select="*" mode="subfile">
      <xsl:with-param name="arg_params" select="$arg_params" />
    </xsl:apply-templates>
  </xsl:copy>
</xsl:template>

<!-- ***** -->
<!-- convert -->
<!-- ***** -->
<xsl:template name="convert">
  <xsl:param name="arg_params" />
  <xsl:param name="arg_att_val" />

  <xsl:choose>
    <xsl:when test="$arg_params">
      <xsl:variable name="para_name" select="$arg_params[1]/@name" />
      <xsl:variable name="para_val" select="$ar

```

```

g_params[1]/@val" />
    <xsl:variable name="conv_att">
      <xsl:call-template name="replace-string">
        <xsl:with-param name="text" select="$arg_att_val" />
        <xsl:with-param name="replace" select="$para_name" />
        <xsl:with-param name="with" select="$para_val" />
      </xsl:call-template>
    </xsl:variable>

    <xsl:call-template name="convert">
      <xsl:with-param name="arg_params" select="$arg_params[position() > 1]" />
      <xsl:with-param name="arg_att_val" select="$conv_att" />
    </xsl:call-template>
  </xsl:when>
  <xsl:otherwise>
    <xsl:value-of select="$arg_att_val" />
  </xsl:otherwise>
</xsl:choose>
</xsl:template>

<!-- ***** -->
      text 中の replace 文字を with 文字で置換する
      ***** -->
<xsl:template name="replace-string">
  <xsl:param name="text" />
  <xsl:param name="replace" />
  <xsl:param name="with" />
  <xsl:choose>
    <xsl:when test="contains($text,$replace)">
      <xsl:value-of select="substring-before($text,$replace)" />
      <xsl:value-of select="$with" />
      <xsl:call-template name="replace-string">
        <xsl:with-param name="text" select="substring-after($text,$replace)" />
        <xsl:with-param name="replace" select="$replace" />
        <xsl:with-param name="with" select="$with" />
      </xsl:call-template>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select="$text" />
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
</xsl:stylesheet>

```

付録 I map_jpn_fo.xsl

```
<?xml version="1.0" encoding="Shift_JIS"?>
<xsl:stylesheet version="1.0"
xmlns:saxon="http://saxon.sf.net/"
xmlns:fn="http://www.w3.org/2005/xpath-functions"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:fo="http://www.w3.org/1999/XSL/Format"
xmlns:axf="http://www.antennahouse.com/names/XSL/Extensions"
xmlns:common="http://exslt.org/common"
exclude-result-prefixes="saxon fn common">

<xsl:output method="xml" indent="yes" encoding="Shift_JIS" />
<xsl:strip-space elements="*" />

<xsl:param name="doc_file" />

<!-- Read xml -->
<xsl:variable name="docfile" select="document($doc_file)" />

<xsl:include href="inc_xsl_fo/comm_val_set.xsl" />
<xsl:include href="inc_xsl_fo/att_set.xsl" />

<!-- Content Transform-->
<xsl:template match="Map">
<fo:root axf: Kerning-mode="contextual" xml:lang="ja" script="ja">
  <fo:layout-master-set>
<!-- === Standard-PageMaster-odd page=== -->
    <fo:simple-page-master
      margin-top="3mm" margin-bottom="3mm" margin-left="20mm" margin-right="12mm" master-name="Standard-PageMaster-ODD"
      page-height="{ $paper-height}"
      page-width="{ $paper-width}">

      <fo:region-body
        margin-top="12mm" margin-bottom="7mm" margin-left="0mm" margin-right="0mm" />
      <fo:region-before extent="10mm" border-bottom-style="solid" border-width="0.1mm" region-name="ODD-Header" />
      <fo:region-after extent="7mm" border-top-style="solid" border-width="0.1mm" region-name="ODD-Footer"/>
    </fo:simple-page-master>

<!-- === First-PageMaster-odd page=== -->
    <fo:simple-page-master
      margin-top="10mm" margin-bottom="3mm" margin-left="20mm" margin-right="12mm" master-name="First-PageMaster-ODD"
      page-height="{ $paper-height}"
      page-width="{ $paper-width}">
```

```
      <fo:region-body
        margin-top="7.5mm" margin-bottom="7mm" margin-left="0mm" margin-right="0mm" />
      <fo:region-before extent="7.5mm" region-name="First-ODD-Header"/>
      <fo:region-after extent="7mm" border-top-style="solid" border-width="0.1mm" region-name="First-ODD-Footer"/>
    </fo:simple-page-master>

<!-- === Standard-PageMaster-even page=== -->
    <fo:simple-page-master
      margin-top="3mm" margin-bottom="3mm" margin-left="12mm" margin-right="20mm" master-name="Standard-PageMaster-EVEN"
      page-height="{ $paper-height}"
      page-width="{ $paper-width}">

      <fo:region-body
        margin-top="12mm" margin-bottom="7mm" margin-left="0mm" margin-right="0mm" />
      <fo:region-before extent="10mm" border-bottom-style="solid" border-width="0.1mm" region-name="EVEN-Header" />
      <fo:region-after extent="7mm" border-top-style="solid" border-width="0.1mm" region-name="EVEN-Footer"/>
    </fo:simple-page-master>

<!-- === First-PageMaster-even page=== -->
    <fo:simple-page-master
      margin-top="10mm" margin-bottom="3mm" margin-left="12mm" margin-right="20mm" master-name="First-PageMaster-EVEN"
      page-height="{ $paper-height}"
      page-width="{ $paper-width}">

      <fo:region-body
        margin-top="7.5mm" margin-bottom="7mm" margin-left="0mm" margin-right="0mm" />
      <fo:region-before extent="7.5mm" region-name="First-EVEN-Header"/>
      <fo:region-after extent="7mm" border-top-style="solid" border-width="0.1mm" region-name="First-EVEN-Footer"/>
    </fo:simple-page-master>

<!-- Set up the sequence of Standard-PageMaster-pages -->
<fo:page-sequence-master master-name="contents">
  <fo:repeatable-page-master-alternatives maximum-repeats="1">
    <fo:conditional-page-master-reference
      master-reference="First-PageMaster-ODD"
      odd-or-even="odd"/>
    <fo:conditional-page-master-reference
      master-reference="First-PageMaster-EVEN">
```



```

        odd-or-even="even"/>
    </fo:repeatable-page-master-alternatives>
    <fo:repeatable-page-master-alternatives>
        <fo:conditional-page-master-reference
            master-reference="Standard-PageMaster
-ODD"
            odd-or-even="odd"/>
        <fo:conditional-page-master-reference
            master-reference="Standard-PageMaster
-EVEN"
            odd-or-even="even"/>
    </fo:repeatable-page-master-alternatives>
</fo:page-sequence-master>
</fo:layout-master-set>

<xsl:call-template name="ResionSet" />
</fo:root>
</xsl:template>

<xsl:template name="ResionSet">
<xsl:variable name="h2_name" select="$docfil
e/SECT2/H2/RBW-ANCHOR/@ID" />

<xsl:variable name="h2_header1">
    <xsl:value-of select="document('header_dat
a_jpn.xml')/TITLES/TITLE[@FROM=$h2_name]/@ID
1" />
</xsl:variable>

<xsl:variable name="h2_header2">
    <xsl:value-of select="document('header_dat
a_jpn.xml')/TITLES/TITLE[@FROM=$h2_name]/@ID
2" />
</xsl:variable>

<xsl:variable name="ismark_width" select="st
ring(Topic[@class='ismark']/@width)" />
<xsl:variable name="ismark_height" select="s
tring(Topic[@class='ismark']/@height)" />

<fo:page-sequence master-reference="contents
">
<fo:static-content flow-name="First-ODD-Head
er">
<fo:block text-align="end"><fo:inline><fo:le
ader />
<xsl:apply-templates select="$docfile/SECT2/
FIG[ISMARK]" mode="top" >
    <xsl:with-param name = "arg_ismark_width" s
elect="$ismark_width" />
    <xsl:with-param name = "arg_ismark_height"
select="$ismark_height" />
</xsl:apply-templates>
</fo:inline></fo:block>
</fo:static-content>

<fo:static-content flow-name="First-ODD-Foot
er">
<fo:block xsl:use-attribute-sets="ODD_FOOTER
"><fo:inline font-size="{Q13}" xml:lang="en

```

```

" script="en"><xsl:value-of disable-output-e
scaping="yes" select="$copyright" /></fo:inl
ine><fo:leader /><fo:page-number/></fo:block
>
</fo:static-content>

<fo:static-content flow-name="First-EVEN-Hea
der">
<fo:block text-align="end"><fo:inline><fo:le
ader />
<xsl:apply-templates select="$docfile/SECT2/
FIG[ISMARK]" mode="top" >
    <xsl:with-param name = "arg_ismark_width" s
elect="$ismark_width" />
    <xsl:with-param name = "arg_ismark_height"
select="$ismark_height" />
</xsl:apply-templates>
</fo:inline></fo:block>
</fo:static-content>

<fo:static-content flow-name="First-EVEN-Foo
ter">
<fo:block xsl:use-attribute-sets="EVEN_FOOTE
R"><fo:page-number/><fo:leader /><fo:inline
font-size="{Q13}" xml:lang="en" script="en"
"><xsl:value-of disable-output-escaping="yes
" select="$copyright" /></fo:inline></fo:blo
ck>
</fo:static-content>

<fo:static-content flow-name="ODD-Header">
<fo:block xsl:use-attribute-sets="H2_HEADER_
ODD" space-before="3.5mm" axf:kerning-mode="
contextual" axf:punctuation-spacing="0em" co
lor="{header_color}"><xsl:value-of select="
$h2_header1" /><fo:inline font-size="{Q18}"
"><xsl:value-of select="$h2_header2" /></fo:i
nline></fo:block>
</fo:static-content>

<fo:static-content flow-name="EVEN-Header">
<fo:block xsl:use-attribute-sets="H2_HEADER_
EVEN" space-before="3.5mm" axf:kerning-mode=
"contextual" axf:punctuation-spacing="0em" c
olor="{header_color}"><xsl:value-of select=
"$h2_header1" /><fo:inline font-size="{Q18}"
"><xsl:value-of select="$h2_header2" /></fo:
inline></fo:block>
</fo:static-content>

<fo:static-content flow-name="ODD-Footer">
<fo:block xsl:use-attribute-sets="ODD_FOOTER
"><fo:inline font-size="{Q13}" xml:lang="en
" script="en"><xsl:value-of disable-output-e
scaping="yes" select="$copyright" /></fo:inl
ine><fo:leader /><fo:page-number/></fo:block
>
</fo:static-content>

<fo:static-content flow-name="EVEN-Footer">

```

```

<fo:block xsl:use-attribute-sets="EVEN_FOOTER"><fo:page-number/><fo:leader /><fo:inline
font-size="{ $Q13}" xml:lang="en" script="en"
><xsl:value-of disable-output-escaping="yes"
select="$copyright" /></fo:inline></fo:block>
</fo:static-content>

<fo:flow flow-name="xsl-region-body" xsl:use-
attribute-sets="HONBUN">
<xsl:apply-templates />
</fo:flow>
</fo:page-sequence>
</xsl:template>

<!-- ++ class="TopTitle" ++
option : pl_font-size, pl_line-height, pl_font-stretch
option : h2_font-size, h2_line-height, h2_font-stretch
option : pl_ref, h2_ref (pl_ref="ref\txd_pl.fo" )
PL, H2 部分の参照ファイル pl h2 の属性を無視してこのファイルを読み込む
-->
<xsl:template match="Topic[@class='TopTitle']">
<xsl:variable name="pl_font-size" select="string(@pl_font-size)" />
<xsl:variable name="pl_line-height" select="string(@pl_line-height)" />
<xsl:variable name="pl_font-stretch" select="string(@pl_font-stretch)" />
<xsl:variable name="h2_font-size" select="string(@h2_font-size)" />
<xsl:variable name="h2_line-height" select="string(@h2_line-height)" />
<xsl:variable name="h2_font-stretch" select="string(@h2_font-stretch)" />

<xsl:variable name="pl_ref" select="string(@pl_ref)" />
<xsl:variable name="h2_ref" select="string(@h2_ref)" />

<fo:block>
<fo:table-and-caption keep-with-next.within-page="always">
<fo:table table-layout="fixed" inline-progression-dimension="{ $max-width}">
<fo:table-column column-width="53mm" column-number="1" />
<fo:table-column column-width="67mm" column-number="2" />
<fo:table-column column-width="58mm" column-number="3" />
<fo:table-body start-indent="0pt" end-indent="0pt" space-before="0pt" space-after="0pt">
<fo:table-row height="20mm">
<fo:table-cell xsl:use-attribute-sets="title.table">

```

```

<fo:block><fo:external-graphic src="figures/pagelogo.jpg" content-width="52.9mm" content-height="19.9mm"/></fo:block>
<!-- <fo:block><fo:external-graphic src="figures/P35S.eps" /></fo:block> -->

</fo:table-cell>
<fo:table-cell xsl:use-attribute-sets="title.table.PL" background-color="{ $pl_bgcolor}">
<xsl:choose>
<xsl:when test="$pl_ref != ''">
<xsl:copy-of select="document($pl_ref)/*" />
</xsl:when>
<xsl:otherwise>
<fo:block xsl:use-attribute-sets="P.L">
<xsl:if test="$pl_font-stretch != ''">
<xsl:attribute name="font-stretch"><xsl:value-of select="$pl_font-stretch" />
</xsl:attribute>
</xsl:if>
<xsl:if test="$pl_font-size != ''">
<xsl:attribute name="font-size"><xsl:value-of select="$pl_font-size" /></xsl:attribute>
</xsl:if>
<xsl:if test="$pl_line-height != ''">
<xsl:attribute name="line-height"><xsl:value-of select="$pl_line-height" /></xsl:attribute>
</xsl:if>
<xsl:apply-templates select="$docfile/SECT2/P.L" />
</fo:block>
</xsl:otherwise>
</xsl:choose>
</fo:table-cell>
<fo:table-cell xsl:use-attribute-sets="title.table.H2" axf: Kerning-mode="contextual" axf:punctuation-spacing="0.4em" background-color="{ $h2_bgcolor}" color="white">
<xsl:choose>
<xsl:when test="$h2_ref != ''">
<xsl:copy-of select="document($h2_ref)/*" />
</xsl:when>
<xsl:otherwise>
<fo:block xsl:use-attribute-sets="H2">
<xsl:if test="$h2_font-stretch != ''">
<xsl:attribute name="font-stretch"><xsl:value-of select="$h2_font-stretch" />
</xsl:attribute>
</xsl:if>
<xsl:if test="$h2_font-size != ''">
<xsl:attribute name="font-size"><

```

```

xsl:value-of select="$h2_font-size" /></xsl:
attribute>
    </xsl:if>
    <xsl:if test="$h2_line-height != ''
">
        <xsl:attribute name="line-height"
><xsl:value-of select="$h2_line-height" /></
xsl:attribute>
        </xsl:if>
        <xsl:apply-templates select="$doc
file/SECT2/H2">
            <xsl:with-param name = "arg_fon
t-size" select="$h2_font-size" />
            </xsl:apply-templates>
        </fo:block>
        </xsl:otherwise>
        </xsl:choose>
    </fo:table-cell>
</fo:table-row>
</fo:table-body>
</fo:table>
</fo:table-and-caption>
</fo:block>
</xsl:template>

<!--      class="include"
    <Topic class="include" href="ref/rohs.fo
" />
-->
<xsl:template match="Topic[@class='include']
">
    <xsl:variable name="href" select="string(@hr
ef)" />

    <xsl:if test="$href != ''">
        <xsl:copy-of select="document($href)/*" />
    </xsl:if>
</xsl:template>

<!--      class="h3"
    <Topic class="h3" href="SECT2/SECT3/H3" /
>
-->
<xsl:template match="Topic[@class='h3']">
    <xsl:variable name="href_name"      select="@h
ref" />
    <xsl:variable name="exp" select="concat('doc
ument(',$ap,$doc_file, $ap, ')', $href_name
)" />

    <xsl:apply-templates select="saxon:evaluate(
$exp)">
        <xsl:with-param name = "arg_span" select="s
tring(@span)" />
        <xsl:with-param name = "arg_font-stretch" s
elect="string(@font-stretch)" />
        <xsl:with-param name = "arg_font-size" sele
ct="string(@font-size)" />
        <xsl:with-param name = "arg_line-height" se
lect="string(@line-height)" />
        <xsl:with-param name = "arg_break-type" sel
ect="string(@break-type)" />

```

```

</xsl:apply-templates>
</xsl:template>

<!--      class="sect4"
    <Topic class="h3" href="SECT2/SECT4[1]" /
>
-->
<xsl:template match="Topic[@class='sect4']">
    <xsl:variable name="href_name"      select="@h
ref" />
    <xsl:variable name="exp" select="concat('doc
ument(',$ap,$doc_file, $ap, ')', $href_name
)" />
    <xsl:apply-templates select="saxon:evaluate(
$exp)" />
</xsl:template>

<!--      class="sect5"
    <Topic class="h3" href="SECT2/SECT4[1]/SE
CT5" />
-->
<xsl:template match="Topic[@class='sect5']">
    <xsl:variable name="href_name"      select="@h
ref" />
    <xsl:variable name="exp" select="concat('doc
ument(',$ap,$doc_file, $ap, ')', $href_name
)" />
    <xsl:apply-templates select="saxon:evaluate(
$exp)" />
</xsl:template>

<!--      class="h4"
    <Topic class="h4" href="SECT2/SECT4[1]/H4
" />
-->
<xsl:template match="Topic[@class='h4']">
    <xsl:variable name="href_name"      select="@h
ref" />
    <xsl:variable name="exp" select="concat('doc
ument(',$ap,$doc_file, $ap, ')', $href_name
)" />
    <xsl:apply-templates select="saxon:evaluate(
$exp)">
        <xsl:with-param name = "arg_span" select="s
tring(@span)" />
        <xsl:with-param name = "arg_font-stretch" s
elect="string(@font-stretch)" />
        <xsl:with-param name = "arg_font-size" sele
ct="string(@font-size)" />
        <xsl:with-param name = "arg_line-height" se
lect="string(@line-height)" />
        <xsl:with-param name = "arg_break-type" sel
ect="string(@break-type)" />
        <xsl:with-param name = "arg_unit-type" sele
ct="string(@unit-type)" />
        <xsl:with-param name = "arg_mmk" select="st
ring(@mmk)" />
        <xsl:with-param name = "arg_ink" select="st
ring(@ink)" />
    </xsl:apply-templates>

```

```

</xsl:template>

<!--          class="h5"
    <Topic class="h5" href="SECT2/SECT4[1]/SECT5[1]/H5" />
    type = "list" l_num="2"
-->
<xsl:template match="Topic[@class='h5']">
<xsl:variable name="href_name"      select="@href" />
<xsl:variable name="exp" select="concat('document(',$ap,$doc_file, $ap, ')', $href_name)" />

<xsl:choose>
    <xsl:when test="@type='list'">
        <xsl:apply-templates select="saxon:evaluate($exp)" mode="list">
            <xsl:with-param name = "arg_span" select="string(@span)" />
            <xsl:with-param name = "arg_font-stretch" select="string(@font-stretch)" />
            <xsl:with-param name = "arg_font-size" select="string(@font-size)" />
            <xsl:with-param name = "arg_line-height" select="string(@line-height)" />
            <xsl:with-param name = "arg_break-type" select="string(@break-type)" />
            <xsl:with-param name = "arg_l_number" select="string(@l_number)" />
        </xsl:apply-templates>
    </xsl:when>
    <xsl:otherwise>
        <xsl:apply-templates select="saxon:evaluate($exp)">
            <xsl:with-param name = "arg_span" select="string(@span)" />
            <xsl:with-param name = "arg_font-stretch" select="string(@font-stretch)" />
            <xsl:with-param name = "arg_font-size" select="string(@font-size)" />
            <xsl:with-param name = "arg_line-height" select="string(@line-height)" />
            <xsl:with-param name = "arg_break-type" select="string(@break-type)" />
            <xsl:with-param name = "arg_unit-type" select="string(@unit-type)" />
            <xsl:with-param name = "arg_mmk" select="string(@mmk)" />
            <xsl:with-param name = "arg_ink" select="string(@ink)" />
        </xsl:apply-templates>
    </xsl:otherwise>
</xsl:choose>
</xsl:template>

<!--          class="h6"
    <Topic class="h6" href="SECT2/SECT4[1]/SECT5[1]/SECT6[1]/H6" />
-->
<xsl:template match="Topic[@class='h6']">

```

```

<xsl:variable name="href_name"      select="@href" />
<xsl:variable name="exp" select="concat('document(',$ap,$doc_file, $ap, ')', $href_name)" />
<xsl:apply-templates select="saxon:evaluate($exp)">
    <xsl:with-param name = "arg_span" select="string(@span)" />
    <xsl:with-param name = "arg_font-stretch" select="string(@font-stretch)" />
    <xsl:with-param name = "arg_font-size" select="string(@font-size)" />
    <xsl:with-param name = "arg_line-height" select="string(@line-height)" />
    <xsl:with-param name = "arg_break-type" select="string(@break-type)" />
    <xsl:with-param name = "arg_unit-type" select="string(@unit-type)" />
    <xsl:with-param name = "arg_mmk" select="string(@mmk)" />
    <xsl:with-param name = "arg_ink" select="string(@ink)" />
</xsl:apply-templates>
</xsl:template>

<!--          class="h7"
    <Topic class="h7" href="SECT2/SECT4[1]/SECT5[1]/SECT6[1]/SECT7[1]/H7" />
    break-type = "page column"
-->
<xsl:template match="Topic[@class='h7']">
<xsl:variable name="href_name"      select="@href" />
<xsl:variable name="exp" select="concat('document(',$ap,$doc_file, $ap, ')', $href_name)" />
<xsl:apply-templates select="saxon:evaluate($exp)">
    <xsl:with-param name = "arg_span" select="string(@span)" />
    <xsl:with-param name = "arg_font-stretch" select="string(@font-stretch)" />
    <xsl:with-param name = "arg_font-size" select="string(@font-size)" />
    <xsl:with-param name = "arg_line-height" select="string(@line-height)" />
    <xsl:with-param name = "arg_break-type" select="string(@break-type)" />
    <xsl:with-param name = "arg_unit-type" select="string(@unit-type)" />
    <xsl:with-param name = "arg_mmk" select="string(@mmk)" />
    <xsl:with-param name = "arg_ink" select="string(@ink)" />
</xsl:apply-templates>
</xsl:template>

<!--          class="main_p"
    <Topic class="main_p" href="SECT2/SECT4[1]/P

```

```

" />
option : type="list" l_number="2"
-->
<xsl:template match="Topic[@class='main_p']">
<xsl:variable name="href_name" select="@href" />
<xsl:variable name="exp" select="concat('document('$ap,$doc_file, $ap, ')', $href_name)" />

<xsl:choose>
  <xsl:when test="@type='list'">
    <xsl:apply-templates select="saxon:evaluate($exp)" mode="list">
      <xsl:with-param name="arg_font-stretch" select="string(@font-stretch)" />
      <xsl:with-param name="arg_font-size" select="string(@font-size)" />
      <xsl:with-param name="arg_line-height" select="string(@line-height)" />
      <xsl:with-param name="arg_l_number" select="string(@l_number)" />
    </xsl:apply-templates>
  </xsl:when>
  <xsl:otherwise>
    <xsl:apply-templates select="saxon:evaluate($exp)">
      <xsl:with-param name="arg_font-stretch" select="string(@font-stretch)" />
      <xsl:with-param name="arg_font-size" select="string(@font-size)" />
      <xsl:with-param name="arg_line-height" select="string(@line-height)" />
    </xsl:apply-templates>
  </xsl:otherwise>
</xsl:choose>
</xsl:template>

<!-- class="att_p"
<Topic class="att_p" href="SECT2/SECT4[1]/P" />
-->
<xsl:template match="Topic[@class='att_p']">
<xsl:variable name="href_name" select="@href" />
<xsl:variable name="exp" select="concat('document('$ap,$doc_file, $ap, ')', $href_name)" />
<xsl:apply-templates select="saxon:evaluate($exp)" mode="att_p">
  <xsl:with-param name="arg_font-stretch" select="string(@font-stretch)" />
  <xsl:with-param name="arg_font-size" select="string(@font-size)" />
  <xsl:with-param name="arg_line-height" select="string(@line-height)" />
</xsl:apply-templates>
</xsl:template>

<!-- class="break"

```

```

<Topic class="break" type="page" />
<Topic class="break" type="column" />
-->
<xsl:template match="Topic[@class='break']">
<xsl:variable name="break" select="@type" />
<fo:block break-before="{ $break }"></fo:block>
</xsl:template>

<!-- class="dan"
  <Topic class="dan" column-count="2" column-gap="6mm" >
    <Topic class="dan" column-count="3" column-gap="6mm" >
-->
<xsl:template match="Topic[@class='dan']">
<xsl:variable name="column-count" select="@column-count" />
<xsl:variable name="column-gap" select="@column-gap" />

<fo:block-container axf:column-count="{ $column-count }" axf:column-gap="{ $column-gap }">
<xsl:apply-templates />
</fo:block-container>
</xsl:template>

<!-- class="block"
  <Topic class="block" width="90mm" height="4mm" />
  縦 4mm, 横 90mm の空白領域を作る
-->
<xsl:template match="Topic[@class='block']">
<xsl:variable name="width" select="@width" />
<xsl:variable name="height" select="@height" />

<fo:block-container inline-progression-dimension="{ $width }" block-progression-dimension="{ $height }">
<fo:block></fo:block>
</fo:block-container>
</xsl:template>

<!-- <Topic class="hr" />
  <Topic class="hr" aki_before="1mm" width="50mm" />
  default Val : spce-before=2mm, spce-after=2mm, leader-length=100%
-->
<xsl:template match="Topic[@class='hr']">
<xsl:variable name="aki_before">
  <xsl:choose>
    <xsl:when test="@aki_before"><xsl:value-of select="@aki_before" /></xsl:when>
    <xsl:otherwise>2mm</xsl:otherwise>
  </xsl:choose>
</xsl:variable>

<xsl:variable name="aki_after">
  <xsl:choose>

```

```

    <xsl:when test="@aki_after"><xsl:value-of
f select="@aki_after" /></xsl:when>
    <xsl:otherwise>2mm</xsl:otherwise>
</xsl:choose>
</xsl:variable>

<xsl:variable name="hr_width">
    <xsl:choose>
        <xsl:when test="@width"><xsl:value-of sel
ect="@width" /></xsl:when>
        <xsl:otherwise>100%</xsl:otherwise>
    </xsl:choose>
</xsl:variable>

    <fo:block font-size="0mm" space-before="{
$aki_before}" space-before.precedence="force
" space-after="{ $aki_after}" space-after.pre
cedence="force" clear="both">
<fo:leader leader-length="{ $hr_width}" leade
r-pattern="rule" rule-style="solid" rule-thi
ckness="0.1mm" color="black" />
    </fo:block>
</xsl:template>

<!-- class="disp_abs_position"
(画像等の絶対位置での表示)
例)
<Topic class="disp_abs_position" href="path
" odd_or_even="odd" top="50mm" left="90mm" w
idth="50mm" height="20mm" />
1234.gif
-->
<xsl:template match="Topic[@class='disp_abs_
position']">
    <xsl:variable name="href_name"      select="@h
ref" />
    <xsl:variable name="exp" select="concat('doc
ument('$ap,$doc_file, $ap, ' )' , $href_name
)" />

    <fo:block-container absolute-position="fixed
">
    <xsl:variable name="left1" select="substring
-before(@left,'mm') " />
    <xsl:variable name="left_val">
    <xsl:choose>
        <xsl:when test="@odd_or_even='odd'">
            <xsl:value-of select="concat($left1 + 17,
'mm') " />
        </xsl:when>
        <xsl:when test="@odd_or_even='even'">
            <xsl:value-of select="concat($left1 + 7, '
mm') " />
        </xsl:when>
    </xsl:choose>
    </xsl:variable>

    <xsl:attribute name="left"><xsl:value-of sel
ect="$left_val" /></xsl:attribute>
    <xsl:if test="@top">
        <xsl:attribute name="top"><xsl:value-of se

```

```

lect="@top" /></xsl:attribute>
    </xsl:if>
    <xsl:if test="@Wwidth">
        <xsl:attribute name="inline-progression-di
mension"><xsl:value-of select="@width" /></x
sl:attribute>
    </xsl:if>
    <xsl:if test="@height">
        <xsl:attribute name="block-progression-dim
ension"><xsl:value-of select="@height" /></x
sl:attribute>
    </xsl:if>
    <xsl:apply-templates select="saxon:evaluat
e($exp)" />
</fo:block-container>
</xsl:template>

<!-- **** class="" table ****
    <Topic class="table" frame="none" width="
90mm">
-->
<xsl:template match="Topic[@class='table']">
<fo:block keep-together.within-page="always"
>
    <fo:table-and-caption>
        <fo:table table-layout="fixed">
            <xsl:if test="@width">
                <xsl:attribute name="inline-progressi
on-dimension">
                    <xsl:value-of select="@width"/>
                </xsl:attribute>
            </xsl:if>
            <xsl:apply-templates select="Topic[@cla
ss='colspec']"/>
            <fo:table-body start-indent="0pt" end-i
ndent="0pt">
                <xsl:apply-templates select="Topic[@c
lass='tr']"/>
            </fo:table-body>
        </fo:table>
    </fo:table-and-caption>
</fo:block>
</xsl:template>

<!-- *** class="colspec" ****
    <Topic class="colspec" colnum="1" width="
32mm" />
    @colnum , @width required
-->
<xsl:template match="Topic[@class='colspec']
">
    <fo:table-column>
        <xsl:attribute name="column-number"><xsl:val
ue-of select="@colnum"/></xsl:attribute>
        <xsl:attribute name="column-width"><xsl:valu
e-of select="@width"/></xsl:attribute>
    </fo:table-column>
</xsl:template>

<!-- *** class="" tr ****
    <Topic class="tr" height="35mm">
-->

```



```

<xsl:template match="Topic[@class='tr']">
<fo:table-row>
  <xsl:if test="@height">
    <xsl:attribute name="height">
      <xsl:value-of select="@height"/>
    </xsl:attribute>
  </xsl:if>
  <xsl:apply-templates />
</fo:table-row>
</xsl:template>

<!-- *** class="" td ****
      <Topic class="td" valign="top" align=
"center">
option : valign, align, colspan, rowspan
-->
<xsl:template match="Topic[@class='td']">
<xsl:variable name="valign">
<xsl:choose>
  <xsl:when test="@valign='top'">before</xsl:
:when>
  <xsl:when test="@valign='middle'">center</
xsl:when>
  <xsl:when test="@valign='bottom'">after</x
sl:when>
  <xsl:otherwise>before</xsl:otherwise>
</xsl:choose>
</xsl:variable>

<xsl:variable name="align">
  <xsl:choose>
    <xsl:when test="@align='left'">left</xsl:
:when>
    <xsl:when test="@align='right'">right</x
sl:when>
    <xsl:when test="@align='center'">center<
/xsl:when>
    <xsl:otherwise>left</xsl:otherwise>
  </xsl:choose>
</xsl:variable>

<xsl:variable name="colspan" select="string(
@colspan)" />
<xsl:variable name="rowspan" select="string(
@rowspan)" />

<fo:table-cell display-align="{ $valign}" tex
t-align="{ $align}" border-style="solid" bord
er-width="0.1mm">
<xsl:if test="ancestor::Topic[@class='table'
][1]/@frame='none'">
  <xsl:attribute name="border-width">0pt</xs
l:attribute>
</xsl:if>
<xsl:if test="$colspan != ''">
  <xsl:attribute name="number-columns-spanne
d"><xsl:value-of select="$colspan"/></xsl:at
tribute>
</xsl:if>
<xsl:if test="$rowspan != ''">
  <xsl:attribute name="number-rows-spanned">
<xsl:value-of select="$rowspan" /></xsl:attr

```

```

ibute>
</xsl:if>
  <fo:block>
    <xsl:apply-templates />
  </fo:block>
</fo:table-cell>
</xsl:template>

<!-- class="rbw-table"
<Topic class="rbw-table" href="SECT2/SECT4[1
]/RBW-TABLE" />
option : @aki_before(unit : mm), @aki_after(
unit : mm), @width (unit : mm)
          @frame = "sides", "none"
          @font-size="2.5mm", @line-height="2.
5mm"
          @col_color_pos ="1 3 5"
          @omitheader ="yes" (改ページ後 HEADER
を非表示にする default は表示
-->
<xsl:template match="Topic[@class='rbw-table
']">
<xsl:variable name="href_name" select="@h
ref" />
<xsl:variable name="exp" select="concat('doc
ument(',$ap,$doc_file, $ap, ')', $href_name
)" />
<xsl:apply-templates select="saxon:evaluate(
$exp)">
  <xsl:with-param name = "arg_aki_before" sel
ect="string(@aki_before)" />
  <xsl:with-param name = "arg_aki_after" sele
ct="string(@aki_after)" />
  <xsl:with-param name = "arg_width" select="
string(@width)" />
  <xsl:with-param name = "arg_frame" select="
string(@frame)" />
  <xsl:with-param name = "arg_font-size" sele
ct="string(@font-size)" />
  <xsl:with-param name = "arg_line-height" se
lect="string(@line-height)" />
  <xsl:with-param name = "arg_col_color_pos"
select="string(@col_color_pos)" />
  <xsl:with-param name = "arg_omitheader" sel
ect="string(@omitheader)" />
</xsl:apply-templates>
</xsl:template>

<!-- class="fig" FIG に含まれる全ての要素を変
換
  <Topic class="fig" href="SECT2/FiG[2]" />
option : width(40mm, 90%), height(40mm, 90%
)
          fig.align (left, center, right), f
ig.t.align(left, center, right)
-->
<xsl:template match="Topic[@class='fig']">
<xsl:variable name="href_name" select="@h
ref" />
<xsl:variable name="exp" select="concat('doc
ument(',$ap,$doc_file, $ap, ')', $href_name
)" />

```

```

<xsl:apply-templates select="saxon:evaluate(
$exp)">
  <xsl:with-param name = "arg_width" select="
string(@width)" />
  <xsl:with-param name = "arg_height" select="
string(@height)" />
  <xsl:with-param name = "arg_fig.align" sele
ct="string(@fig.align)" />
  <xsl:with-param name = "arg_fig.t.align" se
lect="string(@fig.t.align)" />
</xsl:apply-templates>
</xsl:template>

<!-- class="fig.t" FIG の FIG.T のみ
  <Topic class="fig.t" href="SECT2/FiG[2]/F
IG.T" />
  option : fig.t.align (left, center,
right) font-stretch (%)
-->
<xsl:template match="Topic[@class='fig.t']">
<xsl:variable name="href_name" select="@h
ref" />
<xsl:variable name="exp" select="concat('doc
ument(',$ap,$doc_file, $ap, ')', $href_name
)" />
<xsl:apply-templates select="saxon:evaluate(
$exp)">
  <xsl:with-param name = "arg_fig.t.align" se
lect="string(@fig.t.align)" />
  <xsl:with-param name = "arg_font-stretch" s
elect="string(@font-stretch)" />
</xsl:apply-templates>
</xsl:template>

<!-- class="fig.p" FIG の P のみ
  <Topic class="fig.p" href="SECT2/FiG[2]/P
" />
-->
<xsl:template match="Topic[@class='fig.p']">
<xsl:variable name="href_name" select="@h
ref" />
<xsl:variable name="exp" select="concat('doc
ument(',$ap,$doc_file, $ap, ')', $href_name
)" />
<xsl:apply-templates select="saxon:evaluate(
$exp)" mode="fig_p" />
</xsl:template>

<!-- class="fig.name" FIG の FIG.NAME のみ
  <Topic class="fig.name" href="SECT2/FiG[2
]/FIG.NAME" />
-->
<xsl:template match="Topic[@class='fig.name'
]">
<xsl:variable name="href_name" select="@h
ref" />
<xsl:variable name="exp" select="concat('doc
ument(',$ap,$doc_file, $ap, ')', $href_name
)" />
<xsl:apply-templates select="saxon:evaluate(
$exp)" mode="fig_name">

```

```

  <xsl:with-param name = "arg_width" select="
string(@width)" />
  <xsl:with-param name = "arg_height" select="
string(@height)" />
  <xsl:with-param name = "arg_fig.align" sele
ct="string(@fig.align)" />
</xsl:apply-templates>
</xsl:template>

<!-- class="att_list_remark"
  <Topic class="att_list_remark" att_list_s
tr ="#" href="SECT2/SECT4[6]/P" />
  option : font-size, line-height
-->
<xsl:template match="Topic[@class='att_list_
remark']">
<xsl:variable name="font-size" select="
string(@font-size)" />
<xsl:variable name="line-height" select="
string(@line-height)" />
<xsl:variable name="href_name" select="@h
ref" />
<xsl:variable name="exp" select="concat('doc
ument(',$ap,$doc_file, $ap, ')', $href_name
)" />
<xsl:variable name="exp_item_numbers" select
="concat('count(',$exp,')')"/>
<xsl:variable name="item_numbers" select="sa
xon:evaluate($exp_item_numbers)" />
<xsl:variable name="st_indent_unit" select="
'1em'" />

<fo:block font-size="{Q10}" line-height="1.
2em" font-family="serif">
<xsl:if test="$font-size != ''">
  <xsl:attribute name="font-size"><xsl:value
-of select="$font-size" /></xsl:attribute>
</xsl:if>
<xsl:if test="$line-height != ''">
  <xsl:attribute name="line-height"><xsl:val
ue-of select="$line-height" /></xsl:attribut
e>
</xsl:if>
<fo:list-block space-after="0mm" provisional
-distance-between-starts="{st_indent_unit}"
provisional-label-separation="0em">
<xsl:apply-templates select="saxon:evaluate(
$exp)" mode="att_list_item_remark">
  <xsl:with-param name = "arg_att_list_str" s
elect="string(@att_list_str)" />
  <xsl:with-param name = "arg_item_numbers" s
elect="$item_numbers" />
</xsl:apply-templates>
</fo:list-block>
</fo:block>
</xsl:template>

<!-- class="att_list_caution"
  <Topic class="att_list_caution" att_list_st
r ="注" href="SECT2/SECT4[6]/P" />
  option : font-size, line-height
-->

```

```

<xsl:template match="Topic[@class='att_list_
caution']">
<xsl:variable name="font-size"      select="
string(@font-size)" />
<xsl:variable name="line-height"    select="
string(@line-height)" />
<xsl:variable name="href_name"      select="@h
ref" />
<xsl:variable name="exp" select="concat('doc
ument(',$ap,$doc_file, $ap, ' ' , $href_name
)'" />

<xsl:variable name="exp_item_numbers" select
="concat('count(',$exp,')')'" />
<xsl:variable name="st_indent_unit" select="
'2em'" />

<fo:block font-size="{Q10}" line-height="1.
2em" font-family="serif">
<xsl:if test="$font-size != ''">
  <xsl:attribute name="font-size"><xsl:value
-of select="$font-size" /></xsl:attribute>
</xsl:if>
<xsl:if test="$line-height != ''">
  <xsl:attribute name="line-height"><xsl:val
ue-of select="$line-height" /></xsl:attribut
e>
</xsl:if>
<fo:list-block space-after="0mm" provisional
-distance-between-starts="{ $st_indent_unit}
" provisional-label-separation="0em">
<xsl:apply-templates select="saxon:evaluate(
$exp)" mode="att_list_item_caution">
  <xsl:with-param name = "arg_att_list_str" s
elect="string(@att_list_str)" />
</xsl:apply-templates>
</fo:list-block>
</fo:block>
</xsl:template>

<!-- class="att_list"
  <Topic class="att_list" att_list_str ="Not
e: " href="SECT2/SECT4[6]/P" />
    option : width (default val = 178mm) font-s
ize, line-height
-->
<xsl:template match="Topic[@class='att_list'
]">
<xsl:variable name="att_list_width" select=
"string(@width)" />
<xsl:variable name="font-size"      select="
string(@font-size)" />
<xsl:variable name="line-height"    select="
string(@line-height)" />

<xsl:variable name="href_name"      select="@h
ref" />
<xsl:variable name="exp" select="concat('doc
ument(',$ap,$doc_file, $ap, ' ' , $href_name
)'" />

<xsl:variable name="exp_item_numbers" select

```

```

="concat('count(',$exp,')')'" />
<xsl:variable name="item_numbers" select="sa
xon:evaluate($exp_item_numbers)" />

<xsl:variable name="att_list_str_len" selec
t="string-length(@att_list_str)" />

<xsl:variable name="coll_width" select="conc
at(round($att_list_str_len div 2),'em')" />

<fo:block font-size="{Q10}" line-height="1.
2em" font-family="serif">
<xsl:if test="$font-size != ''">
  <xsl:attribute name="font-size"><xsl:value
-of select="$font-size" /></xsl:attribute>
</xsl:if>
<xsl:if test="$line-height != ''">
  <xsl:attribute name="line-height"><xsl:val
ue-of select="$line-height" /></xsl:attribut
e>
</xsl:if>

<fo:table-and-caption>
<fo:table table-layout="fixed" space-before=
"2pt" space-after="2pt" inline-progression-d
imension="{ $max-width}">
<xsl:if test="$att_list_width != ''">
  <xsl:attribute name="inline-progression-di
mension"><xsl:value-of select="$att_list_wid
th" /></xsl:attribute>
</xsl:if>
<xsl:choose>
  <xsl:when test="$item_numbers=1">
    <fo:table-column column-number="1" column
-width="{ $coll_width}" text-align="left" dis
play-align="before"/>
    <fo:table-column column-number="2" text-a
lign="left" display-align="before"/>
  </xsl:when>
  <xsl:otherwise>
    <fo:table-column column-number="1" column
-width="{ $coll_width}" text-align="left" dis
play-align="before"/>
    <fo:table-column column-number="2" column
-width="1em" text-align="left" display-align
="before"/>
    <fo:table-column column-number="3" text-a
lign="left" display-align="before"/>
  </xsl:otherwise>
</xsl:choose>
<fo:table-body start-indent="0pt" end-indent
="0pt">
<xsl:apply-templates select="saxon:evaluate(
$exp)" mode="att_list_item">
  <xsl:with-param name = "arg_att_list_str" s
elect="string(@att_list_str)" />
  <xsl:with-param name = "arg_item_numbers" s
elect="$item_numbers" />
</xsl:apply-templates>
</fo:table-body>
</fo:table>
</fo:table-and-caption>

```

```

</fo:block>
</xsl:template>

<!-- class="float"
    <Topic class="float" href="SECT2/FIG[2]"
    width="96mm" aki="6mm" />
    option : height (40mm, 90%), aki(40mm, 90%)
-->
<xsl:template match="Topic[@class='float']">
<xsl:variable name="href_name" select="@href" />
<xsl:variable name="exp" select="concat('document(',$ap,$doc_file, $ap, ')', $href_name
)' " />

<fo:float float="start">
    <fo:block-container>
        <xsl:attribute name="inline-progression-
dimension"><xsl:value-of select="@width"/></
xsl:attribute>
        <xsl:if test="@height">
            <xsl:attribute name="block-progression-
dimension"><xsl:value-of select="@height"/><
/xsl:attribute>
        </xsl:if>
        <fo:block>
            <xsl:if test="@style='start'">
                <xsl:attribute name="end-indent"><xsl
:value-of select="@aki"/></xsl:attribute>
            </xsl:if>
            <xsl:choose>
                <xsl:when test="@href">
                    <xsl:apply-templates select="saxon:
evaluate($exp)" />
                </xsl:when>
                <xsl:otherwise>
                    <xsl:apply-templates />
                </xsl:otherwise>
            </xsl:choose>
        </fo:block>
    </fo:block-container>
</fo:float>
</xsl:template>

<!-- class="float_clear"
    <Topic class="float_clear" />
-->
<xsl:template match="Topic[@class='float_cle
ar']">
<fo:block clear="both">
    <fo:block-container inline-progression-d
imension="100%" block-progression-dimension=
"0.1pt">
        <fo:block></fo:block>
    </fo:block-container>
</fo:block>
</xsl:template>

<xsl:include href="inc_xsl_fo/element_lib.xs
l" />
<xsl:include href="inc_xsl_fo/table.xml" />
<xsl:include href="inc_xsl_fo/fig.xml" />

```

```

<xsl:include href="inc_xsl_fo/inline.xml" />
<xsl:include href="inc_xsl_fo/not_disp.xml"
/>
</xsl:stylesheet>

```

付録J tenkai.sch(マッピングファイル検証用スキマトロンスキーマ)

```
<?xml version="1.0" encoding="utf-8"?>
<schema xmlns="http://www.ascc.net/xml/schematron">
  <pattern name="Basic Tenkai Validator">
    <rule context="/">
      <assert test="Map">
        The root element of a content page must be named
        'Map'.
      </assert>
    </rule>
    <!-- Map must id -->
    <rule context="Map">
      <assert test="@id">
        A Map element must contain a id attribute.
      </assert>
      <assert test="count(*) = count(Topic | Topic_For_each | Topic_Call | TopicIF | Topic_Chose | Topic_Otherwise)">
        Unexpected elements found !
      </assert>
      <assert test="((Topic[1]/@class = 'ismark') and (Topic[2]/@class = 'TopTitle')) or (Topic[1]/@class = 'TopTitle')">
        1st Topic class="ismark" 2nd Topic class="TopTitle" or
        1nd Topic class="TopTitle"
      </assert>
    </rule>

    <!-- Topic[@class='ismark']
           optional width, height -->
    <rule context="Topic[@class='ismark']">
      <assert test="count(@*) = count(@*[local-name()='class'] | @*[local-name()='width'] | @*[local-name()='height'])">
        attributes name class or width or height !
      </assert>
    </rule>

    <!-- Topic[@class='TopTitle']
           optional pl_font-size, pl_line-height, pl_font-stretch
           h2_font-size, h2_line-height, h2_font-stretch, h2_ref
    -->
    <rule context="Topic[@class='TopTitle']">
      <assert test="count(@*) = count(@*[local-name()='class'] | @*[local-name()='pl_font-size'] | @*[local-name()='pl_line-height'] | @*[local-name()='pl_font-stretch'] | @*[local-name()='h2_font-size'] | @*[local-name()='h2_line-height'] | @*[local-name()='h2_font-stretch'] | @*[local-name()='h2_ref'])">
        attributes name class or pl_font-size or pl_line-height or pl_font-stretch or h2_font-size or h2_line-height or h2_font-stretch or h2_ref !
      </assert>
    </rule>

    <!-- Topic[@class='h3']
           must href
           optional
           font-size, line-height, font-stretch, break-type(page or column)
    -->
    <rule context="Topic[@class='h3']">
      <assert test="@href">
        must contains href!
      </assert>
      <report test="@href = ''">
        href attribute is EMPTY!
      </report>

      <assert test="count(@*) = count(@*[local-name()='class'] | @*[local-name()='href'] | @*[local-name()='font-size'] | @*[local-name()='line-height'] | @*[local-name()='font-stretch'] | @*[local-name()='break-type'])">
        attributes name class or font-size or line-height or font-stretch or break-type!
      </assert>

      <report test="(@break-type) and not(@break-type='page' or @break-type='column')">
        @break-type must be 'page' or 'column'.
      </report>
    </rule>

    <!-- Topic[@class='h4']
           must href
           optional
           font-size, line-height, font-stretch, break-type(page or column)
    -->
    <rule context="Topic[@class='h4']">
      <assert test="@href">
        must contains href!
      </assert>
      <report test="@href = ''">
        href attribute is EMPTY!
      </report>

      <assert test="count(@*) = count(@*[local-name()='class'] | @*[local-name()='href'] | @*[local-name()='font-size'] | @*[local-name()='line-height'] | @*[local-name()='font-stretch'] | @*[local-name()='break-type'])">
        attributes name class or font-size or line-height or font-stretch or break-type!
      </assert>

      <report test="(@break-type) and not(@break-type='page' or @break-type='column')">
        @break-type must be 'page' or 'column'.
      </report>
    </rule>
  </pattern>
</schema>
```

```
ch or h2_ref !
  </assert>
</rule>

<!-- Topic[@class='h3']
      must href
      optional
      font-size, line-height, font-stretch, break-type(page or column)
-->
<rule context="Topic[@class='h3']">
  <assert test="@href">
    must contains href!
  </assert>
  <report test="@href = ''">
    href attribute is EMPTY!
  </report>

  <assert test="count(@*) = count(@*[local-name()='class'] | @*[local-name()='href'] | @*[local-name()='font-size'] | @*[local-name()='line-height'] | @*[local-name()='font-stretch'] | @*[local-name()='break-type'])">
    attributes name class or font-size or line-height or font-stretch or break-type!
  </assert>

  <report test="(@break-type) and not(@break-type='page' or @break-type='column')">
    @break-type must be 'page' or 'column'.
  </report>
</rule>

<!-- Topic[@class='h4']
      must href
      optional
      font-size, line-height, font-stretch, break-type(page or column)
-->
<rule context="Topic[@class='h4']">
  <assert test="@href">
    must contains href!
  </assert>
  <report test="@href = ''">
    href attribute is EMPTY!
  </report>

  <assert test="count(@*) = count(@*[local-name()='class'] | @*[local-name()='href'] | @*[local-name()='font-size'] | @*[local-name()='line-height'] | @*[local-name()='font-stretch'] | @*[local-name()='break-type'])">
    attributes name class or font-size or line-height or font-stretch or break-type!
  </assert>

  <report test="(@break-type) and not(@break-type='page' or @break-type='column')">
    @break-type must be 'page' or 'column'.
  </report>
</rule>
```



```

    </assert>
  </rule>

  <!-- Topic[@class='att_list_caution']
  must      href, att_list_str
  optional
           font-size, line-height
-->
  <rule context="Topic[@class='att_list_caution']">
    <assert test="@href">
      must contains href!
    </assert>
    <report test="@href = ''">
      href attribute is EMPTY!
    </report>

    <assert test="@att_list_str">
      must contains att_list_str!
    </assert>
    <report test="@att_list_str = ''">
      att_list_str attribute is EMPTY!
    </report>

    <assert test="count(@*)= count(@*[local-name()='class'] | @*[local-name()='href'] | @*[local-name()='font-size'] | @*[local-name()='line-height'] | @*[local-name()='att_list_str'])">
      attributes name class or font-size or line-height or att_list_str !
    </assert>
  </rule>

  <!-- Topic[@class='att_list']
  must      href, att_list_str
  optional
           font-size, line-height, width
-->
  <rule context="Topic[@class='att_list']">
    <assert test="@href">
      must contains href!
    </assert>
    <report test="@href = ''">
      href attribute is EMPTY!
    </report>

    <assert test="@att_list_str">
      must contains att_list_str!
    </assert>
    <report test="@att_list_str = ''">
      att_list_str attribute is EMPTY!
    </report>

    <assert test="count(@*)= count(@*[local-name()='class'] | @*[local-name()='href'] | @*[local-name()='font-size'] | @*[local-name()='line-height'] | @*[local-name()='att_list_str'] | @*[local-name()='width'])">
      attributes name class or font-size or line-height or att_list_str or width!
  </rule>

```

```

    </assert>
  </rule>

  <!-- Topic[@class='fig']
  must      href
  optional
           width, height, fig.align, fig.t.
align
-->
  <rule context="Topic[@class='fig']">
    <assert test="@href">
      must contains href!
    </assert>
    <report test="@href = ''">
      href attribute is EMPTY!
    </report>

    <assert test="count(@*)= count(@*[local-name()='class'] | @*[local-name()='href'] | @*[local-name()='width'] | @*[local-name()='height'] | @*[local-name()='fig.align'] | @*[local-name()='fig.t.align'])">
      attributes name class or href or width or height or fig.align or fig.t.align!
    </assert>

    <report test="(@fig.align) and not(@fig.align='left' or @fig.align='right' or @fig.align='center')">
      @fig.align must be 'left' or 'right' or 'center'.
    </report>

    <report test="(@fig.t.align) and not(@fig.t.align='left' or @fig.t.align='right' or @fig.t.align='center')">
      @fig.t.align must be 'left' or 'right' or 'center'.
    </report>
  </rule>

  <!-- Topic[@class='fig.t']
  must      href
  optional
           fig.t.align, font-stretch
-->
  <rule context="Topic[@class='fig.t']">
    <assert test="@href">
      must contains href!
    </assert>
    <report test="@href = ''">
      href attribute is EMPTY!
    </report>

    <assert test="count(@*)= count(@*[local-name()='class'] | @*[local-name()='href'] | @*[local-name()='fig.t.align'] | @*[local-name()='font-stretch'])">
      attributes name class or href or fig.t.align or font-stretch!
    </assert>
  </rule>

```

```

    <report test="(@fig.t.align) and not(@fig.t.align='left' or @fig.t.align='right' or @fig.t.align='center')">
        @fig.t.align must be 'left' or 'right' or 'center'.
    </report>
</rule>

<!-- Topic[@class='fig.p']
must href
-->
<rule context="Topic[@class='fig.p']">
    <assert test="@href">
        must contains href!
    </assert>
    <report test="@href = ''">
        href attribute is EMPTY!
    </report>

    <assert test="count(@*)= count(@*[local-name()='class'] | @*[local-name()='href'])">
        attributes name class or href !
    </assert>
</rule>

<!-- Topic[@class='fig.name']
must href
optional
width, height, fig.align
-->
<rule context="Topic[@class='fig.name']">
    <assert test="@href">
        must contains href!
    </assert>
    <report test="@href = ''">
        href attribute is EMPTY!
    </report>

    <assert test="count(@*)= count(@*[local-name()='class'] | @*[local-name()='href'] | @*[local-name()='width'] | @*[local-name()='height'] | @*[local-name()='fig.align'])">
        attributes name class or href or width or height or fig.align !
    </assert>

    <report test="(@fig.align) and not(@fig.align='left' or @fig.align='right' or @fig.align='center')">
        @fig.align must be 'left' or 'right' or 'center'.
    </report>
</rule>

<!-- Topic[@class='rbw-table']
must href
optional
aki_before, aki_after, width, frame, font-size, line-height, col_color_pos, omitheader

```

```

-->
<rule context="Topic[@class='rbw-table']">
    <assert test="@href">
        must contains href!
    </assert>
    <report test="@href = ''">
        href attribute is EMPTY!
    </report>

    <assert test="count(@*)= count(@*[local-name()='class'] | @*[local-name()='href'] | @*[local-name()='width'] | @*[local-name()='aki_before'] | @*[local-name()='aki_after'] | @*[local-name()='frame'] | @*[local-name()='font-size'] | @*[local-name()='line-height'] | @*[local-name()='col_color_pos'] | @*[local-name()='omitheader'])">
        attributes name class or href or width or aki_before or aki_after or frame or font-size or line-height or col_color_pos or omitheader !
    </assert>

    <report test="(@frame) and not(@frame='none' or @frame='sides')">
        @fig.align must be 'none' or 'sides'.
    </report>

    <report test="(@omitheader) and not(@omitheader='yes')">
        @omitheader must be 'yes'.
    </report>
</rule>

<!-- Topic[@class='table']
optional
frame, width
-->
<rule context="Topic[@class='table']">
    <assert test="count(@*)= count(@*[local-name()='class'] | @*[local-name()='width'] | @*[local-name()='frame'])">
        attributes name class or width or frame !
    </assert>

    <report test="(@frame) and not(@frame='none')">
        @frame must be 'none' .
    </report>
</rule>

<!-- Topic[@class='colspec']
must
colnum, width
-->
<rule context="Topic[@class='colspec']">
    <assert test="count(@*)= count(@*[local-name()='class'] | @*[local-name()='width'] | @*[local-name()='colnum'])">

```



```

        attributes name class or width or col
num !
    </assert>

    <assert test="(count(@width)= 1) and (c
ount(@colnum) =1)">
        must contains width and colnum attrib
ute
    </assert>

</rule>

<!-- Topic[@class='tr']
option
    width
-->
<rule context="Topic[@class='tr']">
    <assert test="count(@*)= count(@*[local
-name()='class'] | @*[local-name()='width'])
">
        attributes name class or width !
    </assert>
</rule>

<!-- Topic[@class='td']
option
    align, valign
-->
<rule context="Topic[@class='td']">
    <assert test="count(@*)= count(@*[local
-name()='class'] | @*[local-name()='align']
| @*[local-name()='valign'])">
        attributes name class or align or val
ign !
    </assert>

    <report test="(@align) and not(@align='
left' or @align='center' or @align='right')
">
        @align must be 'left' or 'center' 'ri
ght' .
    </report>

    <report test="(@valign) and not(@valign
='top' or @valign='middle' or @valign='botto
m')">
        @valign must be 'top' or 'middle' or
'bottom' .
    </report>
</rule>

<!-- Topic[@class='block']
must width, height -->
<rule context="Topic[@class='block']">
    <assert test="count(@*)=3">
        must contains 3 attributes (include c
lass attribute) !
    <assert test="(count(@width)= 1) and
(count(@height) =1)">
        must contains width attribute and h
eight attribute
    </assert>

```

```

    </assert>
</rule>

<!-- Topic[@class='hr']
optional width, aki_before, aki_aft
er -->
<rule context="Topic[@class='hr']">
    <assert test="count(@*)= count(@*[local
-name()='class'] | @*[local-name()='width']
| @*[local-name()='aki_before'] | @*[local-
name()='aki_after'])">
        attributes name class or width or aki
_before or aki_after !
    </assert>
</rule>

<!-- Topic[@class='break']
must type -->
<rule context="Topic[@class='break']">
    <assert test="@type">
        must contains type attribute !
    </assert>
    <assert test="(@type='page') or (@type=
'column')">
        must type attribute value is page or
column ! This value is <value-of select="@typ
e"/>.
    </assert>
</rule>

<!-- Topic[@class='include']
must href -->
<rule context="Topic[@class='include']">
    <assert test="@href">
        must contains href attribute !
    </assert>
</rule>

<!-- Topic[@class='dan'] must column-cou
nt, column-gap -->
<rule context="Topic[@class='dan']">
    <assert test="count(@*)=3">
        must contains 3 attributes (include c
lass attribute) !
    <assert test="(count(@column-count)=
1) and (count(@column-gap) =1)">
        must contains column-count attribut
e and column-gap attribute
    </assert>
</rule>

<!-- Topic_For_each
must for, from, to -->
<rule context="Topic_For_each">
    <assert test="(count(@for)= 1) and (cou
nt(@from) =1) and (count(@to) =1)">
        must contains for and from and to att
ribute
    </assert>

```

```

    <assert test="substring(@for,1,1) = '$'
">
    @from not Integer !
    </assert>

    <assert test="floor(@from) = number(@fr
om)">
    @from not Integer !
    </assert>

    <assert test="floor(@to) = number(@to)"
>
    @from not Integer !
    </assert>
</rule>

<!-- TopicIF must test -->
<rule context="TopicIF">
    <assert test="count(@test)= 1">
        must contains test attribute
    </assert>

    <report test="@test = ''">
        @test is not null !
    </report>
</rule>

<!-- Topic_Chose -->
<rule context="Topic_Chose">
    <assert test="count(*) = count(Topic_Wh
en | Topic_Otherwise)">
        element must be Topic_When or Topic_O
therwise !
    </assert>
</rule>

<!-- Topic_When must test -->
<rule context="Topic_When">
    <assert test="Topic">
        element must contain Topic !
    </assert>

    <assert test="count(@test)= 1">
        must contains test attribute
    </assert>

    <report test="@test = ''">
        @test is not null !
    </report>
</rule>

<!-- Topic_Otherwise -->
<rule context="Topic_Otherwise">
    <assert test="Topic">
        element must contain Topic !
    </assert>
</rule>

<!-- Topic_Call must href -->
<rule context="Topic_call">
    <assert test="Topic_With_Param">

```

```

        element must contain Topic_With_Param
!
    </assert>

    <assert test="count(@href)= 1">
        must contains href attribute
    </assert>

    <report test="@href = ''">
        @href is not null !
    </report>
</rule>

<!-- Topic_With_Param must name, val -->
<rule context="Topic_With_Param">
    <assert test="normalize-space(.) = ''">
        element must be EMPTY !
    </assert>

    <assert test="count(@name)= 1">
        must contains name attribute
    </assert>
    <report test="@name = ''">
        @name is not null !
    </report>

    <assert test="count(@val)= 1">
        must contains val attribute
    </assert>
    <report test="@val = ''">
        @val is not null !
    </report>
</rule>
</pattern>
</schema>

```